AD-A099 332 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 20/4
PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLU--ETC(U)
NOV 80 J MULLEN, F K GOODWIN, M F DILLENIUS F33615-76-C-3077
UNCLASSIFIED NEAR-TR-210-VOL-4 AFWAL-TR-80-3032-VOL-4 NL

1 OF 4
AD A
099332

AD A099 ??2

AFWAL-TR-80-3032

VOLUME IV

PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLUDING FUSELAGE AND STORES OF NONCIRCULAR CROSS SECTION. VOLUME IV - APPENDICES C AND D, DETAILS OF PROGRAM II

Joseph Mullen, Jr.
Frederick K. Goodwin
Marnix F. E. Dillenius
Nielsen Engineering & Research, Inc.
Mountain View, California 94043

November 1980

TECHNICAL REPORT AFWAL-TR-80-3032, VOLUME IV
FINAL REPORT FOR PERIOD JUNE 1975 - FEBRUARY 1980

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Calvin L. Dyer

CALVIN L. DYER
Project Engineer

R.O. Anderson

RONALD O. ANDERSON, Chief
Control Dynamics Branch

FOR THE COMMANDER

Robert C. Ettinger

ROBERT C. ETTINGER, Col, USAF, Chief
Flight Control Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIGC, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on the specific document.

(19) TR-80-3032-VOL-4

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFWAL-TR-80-3032, Vol. IV	2. GOVT ACCESSION NO.: AD-A099	3. RECIPIENT'S CATALOG NUMBER: 332
4. TITLE (and Subtitle): PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLUDING FUSELAGE AND STORES OF NONCIRCULAR CROSS SECTION. Volume IV. Appendices C and D, Details of Program II.		5. TYPE OF REPORT & PERIOD COVERED: (9) Final Report. June 1975 — February 1980
		6. PERFORMING ORG. REPORT NUMBER: (14) NEAR-TR-210-VOL-4
7. AUTHOR(s): (10) Joseph Mullen, Jr., Frederick K. Goodwin, Marnix F. E. Dillenius		8. CONTRACT OR GRANT NUMBER(s): (15) F33615-76-C-3077
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Nielsen Engineering & Research, Inc., 510 Clyde Avenue, Mountain View, CA 94043 (12) 327		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project 2403, (16) Task 240305, Work Unit 240309 (17) 05
11. CONTROLLING OFFICE NAME AND ADDRESS: Flight Dynamics Laboratory, Air Force Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE: (11) November 1980
		13. NUMBER OF PAGES: 328
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aerodynamic Loads
Aerodynamic Interference
External Stores
Flow Fields
Store Separation
Supersonic Flow

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Detailed instructions are presented for using a computer program which calculates the six-degree-of-freedom trajectories of external stores which are separated from fighter-bomber type aircraft flying at supersonic speeds. Multiple circular or elliptical store configurations may be handled. Parent aircraft configurations may consist of a circular or arbitrary cross section fuselage with ramp external compression inlets, and a wing, pylon, and rack. The program uses linear potential-flow theory to model the wing and pylon loading and thickness. Three-dimensional line sources and doublets are used

DD FORM 1 JAN 73 1473

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

to model circular fuselages and stores. The noncircular fuselage and elliptic store surfaces are modeled with constant source panels. Nonlinear corrections are made to the wing, fuselage, rack, store and fuselage inlet models to simulate shocks. The program also calculates the trajectory of the store as it separates from the aircraft. This report describes the program, presents instructions for preparing input for the program, describes the output from the program, and presents a sample case. The program represents an extension of an earlier program restricted to circular bodies at supersonic speeds, written by the present authors and described in AFFDL-TR-76-41.

This volume presents the detailed descriptions of the calculations performed in each of the subroutines in Program II. Also included are the descriptions of each of the variables passed between routines.

A

FOREWORD

This report, "Prediction of Supersonic Store Separation Characteristics Including Fuselage and Stores of Noncircular Cross Section," describes a combined theoretical-experimental program directed toward developing a computer program for predicting the trajectory of an external store separated from an aircraft flying at supersonic speed. It represents an extension of previous work covered in AFFDL-TR-76-41 to include more realistic modeling of fuselage shapes including noncircular cross sections and ramp type engine air inlets, and to include modeling store shapes with elliptic cross section with multiple sets of arbitrary oriented fins. Volume I, "Theoretical Methods and Comparisons with Experiment," describes the theoretical approach and presents extensive comparisons with experimental data. Volume II, "Users Manual for the Computer Program," presents detailed instructions on the use of the computer program with emphasis on preparation of input data and interpretation of output. Volume III, "Appendices A and B, Details of Program I," provides additional descriptions of the individual subroutines and program variables passed between modules in the first of two programs. This volume, Volume IV, "Appendices C and D, Details of Program II," provides additional descriptions of the individual subroutines and program variables passed between modules in the second program.

This work was carried out by Nielsen Engineering & Research, Inc., 510 Clyde Avenue, Mountain View, California 94043, under Contract No. F33615-76-C-3077. The contract was initiated under Project 2403, Task 240305, of the Air Force Flight Dynamics Laboratory. The Air Force Project Engineer on the contract was Calvin L. Dyer, AFWAL/FIGC. The report number assigned by Nielsen Engineering & Research, Inc. is NEAR TR 210.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF FIGURES

LIST OF TABLES

PREDICTION OF SUPERSONIC STORE SEPARATION
CHARACTERISTICS INCLUDING FUSELAGE AND
STORES OF NONCIRCULAR CROSS SECTION

Volume IV - Appendices C and D, Details of Program II

SUMMARY

The purpose of this volume is to provide the additional details of the parts of the second program that would be useful to the programmer or engineer interested in understanding the calculations and computer code herein. The information included here as Appendices C and D describes the function and operations performed by each routine, a description of data transferred between subroutines and a listing of Program II itself.

Appendix C provides a detailed description of the operations and flow of calculations of each of the individual routines in the second program. Included are a description of the flow of the calculations, including flow charts of some routines, a description of any program arguments, and a program listing.

Appendix D provides a description of all variables passed between routines in common blocks. A listing of each common block with a description of each variable, array or index in the common are provided. A special section is provided for the multiple uses of blank common as well as a cross reference chart of routine versus common block usage.

APPENDIX C

DETAILS OF PROGRAM II SUBROUTINES

C-1 Introduction

The purpose of this Appendix is to provide more detailed information on the calculations performed in the second program whose use was described in Section 4 of Reference 1, and whose methods are described in Reference 2. A listing of Program II is presented in Figure C-1 and a schematic showing the subroutine calling sequence is presented in Figure C-2. This appendix will present the details of the flow calculations of each of the routines, a description of the variables in the argument list, individual detailed flow charts, and a summary of the functions of each routine. The program consists of a main program and 106 subroutines. The main program will be described and then the subroutines will be described in alphabetical order. The subroutines and their functions are listed in Table C-1. Flow charts of some of the individual routines are shown in Figures C-3 through C-15. Refer to Volume II of this report for the list of symbols used.

TABLE C-1

SUBROUTINES USED IN PROGRAM II

Subroutine Name	Function
TRJTRY	main program to read in data describing separating store, set up store and empennage models, organize calculation of store forces and moments, and set up and solve equations of motion for store trajectory

ADAMS	numerical integration routine to integrate differential equations
BDCOEF	routine to organize generation of influence coefficient matrices for elliptic store body at empennage control points
BDUVW	routine to organize calculation of U,V,W velocities from coefficient matrices computed at empennage control points
BVARIA	calculates betas to be used for axisymmetric or noncircular fuselage, rack, and all circular and elliptic store bodies except the separated store
CEL1	calculates complete elliptic integral of the first kind
CEL2	calculates complete elliptic integral of the second kind
CRFWBD	organizes the reading and printing of the input, setting up of the geometry and calculation of the influence coefficient matrix for multi-fin elliptic-store empennages
DASCRU	adjustable interval integration routine for ordinary linear differential equations
DBLU	calculates the intermediate transform variable w for the conformal transformation of an elliptical body with wings.

DEMON2	forms the right hand side of a multiple fin/interference shell velocity equations in the presence of a nonuniform flow field and calls for the solution for the u-velocity panel strengths, pressures and loads
DIRCOS	calculates direction cosines between inertial and store body coordinate systems
DSDZ	complex transformation used by VVELS in vortex tracking
DZDS	complex transformation used by VOTEX in vortex tracking
D2SDZ2	complex transformation used by VOTEX in vortex tracking
EGMSAV	routine to save elliptic store empennage geometry on TAPE3
EGMRST	routine to restore elliptic store empennage geometry from TAPE3
EJECTR	calculates ejector-forces and moments from input polynomials
ELI1	calculates generalized elliptic integral of the first kind
ELI2	calculates generalized elliptic integral of the second kind

ELLCPT	performs extrapolation of panel control points on ellipse to exact surface along radial line from axis
ELLSHP	computes the horizontal and vertical semi-axes and their axial derivatives for an ellipse
ELRFLB	computes the image elliptic store location relative to the fuselage
ELRFLW	computes the shock shape between the store and wing surface for elliptic stores and calls REFSHK to calculate the image store location
EXPAND	calculates velocities in the cross flow plane of an expanding or contracting elliptical section
F	called by DASCRU to organize calculation of cross flow plane velocities for vortex tracking
FLDAC2	computes influence coefficients for u,v,w velocity components induced by body source panels of a single segment at field points
FLDAIC	organizes calculation of influence coefficients due to body source panels at empennage control points
FLDUVW	computes u,v,w velocities at field points from coefficient matrix generated by FLDAC2 and panel strengths
FLDVEL	organizes computation of u,v,w velocity components at field points

FLDVL2	computes u,v,w velocities at field points due to a single ring of panels
FORMOM	calculates force and moment coefficients on a noncircular body
FREFSH	calculates the location at which the circular store nose shock which is reflected by the fuselage strikes the store and its associated β
FRSTRT	saves or restores required program information for source paneling method to restart configuration analysis
FXBOD	sums the axial distributions of forces and moments for various store-body source panel sections
IMAGEV	computes the influence of a ring of panels on the image elliptic store on the real store
IMAGFN	computes the influence of the image elliptic store body on the real store fins.
IMAGYZ	computes the control point locations of the image store as seen from the body fixed coordinate system of the real elliptic store
IMSVEL	calculates the velocities due to the image circular-body store
INLBET	interpolates in the wedge shaped inlet shock for β of inlet panels used in flow field calculations

INLTST	determines whether a source panel is an inlet panel
INTOST	transforms a vector with components in the inertial coordinate system to one with components in the store body coordinate system
INVER2	solves a system of simultaneous algebraic equations
IOREAD	performs an unformatted read from external file, IO
IOWRIT	performs an unformatted write onto the external file, IO
IXBOD	scans elliptic store geometry to define leading and trailing edges of body-fin sections and elliptic body axes
LAYBIP	lays out and determines geometrical properties of the constant u-velocity panels on the interference shell of the elliptic store body
LAYOUT	lays out and determines geometrical properties of the constant u-velocity panels on the fin surfaces
LOADS	calculates the forces and moments acting on the elliptic store fins and the interference shell
NUMACH	determines the leading edge and trailing edge locations and Mach numbers associated with wing thickness

PANVEL — organizes the calculation of the influence of a single source panel and its image, if a noncircular fuselage panel, on a field point, and the transformation from the local panel coordinate system to the body coordinate system

PAS001 — performs the [L*U] decomposition of a positive definite matrix

PAS002 — solves the system of equations [L*U] * X = B

PITROL — computes the crossflow velocity components for an elliptic body subjected to a nonuniform flow field V_{avg}, W_{avg} along the centerline

PRESS — computes source panel pressure coefficient using exact Bernoulli formula

RDFILE — reads input file produced by Program I which contains all of the data read or computed in Program I and used in Program II

REFSHK — calculates the location at which the circular store nose shock which is reflected by the wing strikes the store and its associated β

RESVEL — computes the resultant velocities at a field point due to all parent aircraft components and stores except the separated store

ROTBW — performs the transformation from the body interference shell panel coordinate system to the empennage coordinate system

ROTFW	performs the transformation from the local fin panel coordinate system to the empennage coordinate system
ROTWB	performs the transformation from the empennage coordinate system to the body interference shell panel coordinate system
ROTWF	performs the transformation from the empennage coordinate system to the local fin panel coordinate system
SDSTN2	organizes the solution for panel strengths and forces and moments for the elliptic store
SDSTRN	calculates the source and doublet strengths for the separated circular store including the effects of the image store
SDTRMS	evaluates the square root and inverse cosh terms appearing in the source and doublet expressions
SEMFOR	calculates the empennage forces and moments for the circular-body store option
SEMPIN	initializes aerodynamic and geometric parameters for the circular-body store empennage model
SFORCE	calculates the aerodynamic forces and moments on the circular-body store using the three dimensional method with line sources and doublets
SFORC2	calculates the aerodynamic forces and moments on an elliptic-body store using the three dimensional method with source panels

SHAPE	calculates the radius and surface slope at a point on a body from input polynomials
SHKLOC	locates the shock wave from an axisymmetric body at coordinates y,z relative to the body
SIMSON	performs Simpson rule integration
SMARCH	solves for the store source panel strengths in the presence of an image body using a ring by ring marching technique
SOLVUV	reads panel on panel influence coefficients from TAPE8 to compute velocity components induced at panel control points
SORPAN	computes the three velocity components induced at a specified control point by a body source panel
SOUTPT	prints the forces, moments, load distributions, and trajectory data at end of each integration step
SPECPR	computes the Bernoulli pressures at control points of the constant u-velocity panels on the fin surfaces
SPNLD	computes the span load distribution for planar or cruciform fin configurations only
STRDAT	reads separated elliptic store data into labeled common and remaining elliptic stores and noncircular fuselage data sequentially in blank common
STTOIN	transforms a vector with components in the store body coordinate system to one with components in the inertial coordinate system

SWINT	finds the intersection of a tabulated store shock wave shape with a circular fuselage
SWINTE	finds the intersection of a tabulated store shock wave shape with a noncircular paneled fuselage
THRCAL	calculates thrust from input polynomials
VELBD2	calculates velocities due to body interference panel corner points
VELCAL	calculates velocities at a field point due to fuselage, rack, or store line sources and doublets
VELDMP	computes the damping velocities due to the rotational velocities of store
VELNOR	calculates the perturbation velocities induced by the empennage u-velocity panels at an empennage panel control point
VELO	calculates the influence of a semi-infinite triangle associated with an empennage u-velocity panel at a field point
VELO2	calculates the influence of a semi-infinite triangle associated with the corner of a constant u-velocity panel on the wing, fuselage, or pylon at a field point
VELOT2	calculates the influence of a semi-infinite triangle associated with a thickness source panel on the wing or pylon at a field point

VELPP2	calculates velocities at a field point due to pylon constant u-velocity panel corner points
VELPT2	calculates velocities at a field point due to pylon thickness panel corner points
VELWP2	calculates velocities at a field point due to wing constant u-velocity panel corner points
VELWT2	calculates velocities at a field point due to wing thickness panel corner points
VNORM	computes the resultant velocity normal to a surface at arbitrary orientations θ and δ from velocities u,v,w
VOTEX	computes the perturbation velocity components at the vortex locations accounting for mutual interference effects and the presence of a finned elliptical body cross section
VPATH	organizes data for calling VPATHL for multi-fin configurations
VPATHL	computes the vortex paths and vortex induced crossflow velocities at specified field points for a set of vortices in the presence of a body at variable angle of attack and roll
VVELS	computes the perturbation velocity components due to NV external vortices and their images inside a body with elliptical cross section
VWAVG	averages the velocity components seen by a ring of body source panels

VXYZ	computes the direction cosines and velocities due to free stream translational motion of the store
XVSR	performs interpolation in a single shock wave shape for X at a given R
XVSRT	performs interpolation in multiple shock wave shapes for X as a function of R and meridional angle
Z	calculates the sigma value in the transformed circle plane for given tau in the physical plane for for elliptical body with fins
ZIMAGE	calculates circular image store velocities at empennage control points
ZIMSVL	calculates the velocities due to the image circular-body store on the real store fins

C-2 Main Program - TRJTRY

The main program TRJTRY has two functions: the initialization of the properties of the separating store and the integration of the time dependent equations of motion. The initialization sequence includes reading the file produced in Program I, reading data defining the properties of the separating store body and fins, and reading data defining optional thrust and ejector forces. The flow chart presented in Figure C-2 shows the basic flow of the calculations. Those pages, the description in Section 3.1 of Reference 1, the input discussion in Section 3.2 of Reference 1, and the descriptions of variables in common in Appendix D should be sufficient information to understand the flow of the program.

The last three pages of the flow chart of Figure 15 of Reference 1 have been expanded and presented in Figure C-3. This

portion of the program begins with FORTRAN statement label 62 of the main program (Figure C-1(c)), and is the integration loop of the program. The first step in the loop is to compute the forces and moments acting on the store body and empennage(s). If the body is circular, the body forces are computed in SFORCE and any empennage forces and moments computed in SEMFOR. The circular model is restricted to one set of fins. If the elliptic body model is used, the body forces are computed in SFORC2 and empennage forces and moments are computed in DEMON2. If two sets of fins exist, routine VPATH may be used for certain configurations to integrate the trajectories of the trailing edge vortices from the first set of fins to the leading edge of the aft set of fins.

When two sets of fins are used under the elliptic body option, an additional computational overhead must be included to compensate for the requirement to use external storage to save the arrays and geometries associated with the panel solutions for the elliptic body and each empennage. During this section of the force and moment calculations blank common must be used to store the arrays used in the solutions for the separating and fixed stores, the noncircular fuselage, and each of the sets of fins. The elliptic store and the noncircular fuselage are stored simultaneously in blank common in the order prescribed by STRDAT. Only the data on TAPE7 is read prior to the elliptic store body solution in SFORC2 and rewritten immediately afterwards to save the new store solution. The first call to EGMRST reinitializes the arrays for the first set of fins and copies the influence coefficient matrices for the first empennage into blank common. After the fin solution TAPE7 is read back into blank common to restore the elliptic body panel strengths for the calculation of the body influence on the aft set of fins. The second call to EGMRST brings the arrays for the second empennage and its influence coefficient matrix into blank common. The solution for the second fin is then computed.

The next portion of the program solves for the translational and rotational accelerations. The set of six simultaneous equations which are solved are given in Appendix B of Reference 2, Equations (B-16) through (B-18) and (B-41) through (B-43). The coefficient matrix and right-hand side are stored in the FVN array. The first subscript of FVN is the equation number. The correspondence is

Subscript Value	Equation Number
1	B-16
2	B-17
3	B-18
4	B-41
5	B-42
6	B-43

The second subscript of FVN is the term in the equation. Here the correspondence is

Subscript Value	Term
1	$\ddot{\xi}$
2	$\ddot{\eta}$
3	$\ddot{\zeta}$
4	$\dot{p}$
5	$\dot{q}$
6	$\dot{r}$
7	Right-hand side

Thus, for example, FVN(3,5) is the coefficient of $\dot{q}$ in Equation (B-18). Certain parts of this section of the program are bypassed

if the store center of gravity, c.g., lies on the store moment center. If this is the case, $\bar{x}$,$\bar{y}$ and $\bar{z}$ are zero and NASYM equals zero.

After calculating the coefficient matrix and right-hand side of the set of equations, subroutine INVER2 is called to solve for the accelerations and the values are transferred to the DVAR array. The next steps of the program put the values of $\dot{\xi}$, $\dot{\eta}$, and $\dot{\zeta}$ into the DVAR array and calculate the values of $\dot{\psi}$, $\dot{\Theta}$, and $\dot{\Phi}$, which are also put in the DVAR array. The DVAR array contains

DVAR(1)	$\ddot{\xi}$
DVAR(2)	$\ddot{\eta}$
DVAR(3)	$\ddot{\zeta}$
DVAR(4)	$\dot{p}$
DVAR(5)	$\dot{q}$
DVAR(6)	$\dot{r}$
DVAR(7)	$\dot{\xi}$
DVAR(8)	$\dot{\eta}$
DVAR(9)	$\dot{\zeta}$
DVAR(10)	$\dot{\psi}$
DVAR(11)	$\dot{\Theta}$
DVAR(12)	$\dot{\Phi}$

These are the derivatives of the twelve dependent variables.

The program next checks to see if the integration procedure has reached the end of an integration step. If it has NOUT=1 and subroutine SOUTPT is called to print the output. Next a check is made to see if the end of the trajectory has been reached, that is, is the current value of the time equal to the final time which was input. If it is then the program stops. If the end has not been reached, then the integration routine, subroutine ADAMS, is called.

NDIFEQ is a control index used by subroutine ADAMS. If NDIFEQ=1 upon returning to the main program an error condition has been encountered in ADAMS and the calculation is to be terminated. If $2 \leq$ NDIFEQ ≤ 7, the program is at some intermediate point in the integration from one point to the next. When NDIFEQ > 7, the integration of one step has been completed and NOUT is set equal to 1 so that the output subroutine will be called after the derivatives are calculated.

Subroutine references:

ADAMS, CRFWBD, DEMON2, DIRCOS, EGMRST, FXBOD, INVER2, IOREAD, IOWRIT, IXBOD, RDFILE, SEMFOR, SEMPIN, SFORCE, SFORC2, SHAPE, SOUTPT, STRDAT, THRCAL, VPATH, VXYZ

C-3 Subroutine ADAMS

Subroutine ADAMS is the subroutine which integrates the set of differential equations. The subroutine will not be described in detail, however, an examination of the flow chart (Figure C-4) and the program listing (Figure C-1(e)) will indicate how it functions. All returns from the subroutine are to the main program.

Consider the following set of n differential equations:

$$\dot{y}_1 = f_1(t, y_1, y_2, \ldots, y_n)$$

$$\vdots$$

$$\dot{y}_n = f_n(t, y_1, y_2, \ldots, y_n)$$

This subroutine uses a fourth-order Adams predictor-corrector method (Reference 3) to solve the above set of equations. To find the value of y_i at the (j+4)th step, the following formula is used to predict the value

$$y_{i,j+4}^{(p)} = y_{i,j+3} + \frac{h}{24} (55\dot{y}_{i,j+3} - 59\dot{y}_{i,j+2} + 37\dot{y}_{i,j+1} - 9\dot{y}_{i,j}) \qquad \text{(C-1)}$$

This assumes that all of the $\dot{y}_i$'s are known for the j, (j+1), (j+2), and (j+3) steps. The quantity h is the interval in the independent variable between these points. After the values of the $y_{i,j+4}^{(p)}$ have been found, the following equation is used to obtain the corrected values:

$$y_{i,j+4} = y_{i,j+3} + \frac{h}{24} (9\dot{y}_{i,j+4}^{(p)} + 19\dot{y}_{i,j+3} - 5\dot{y}_{i,j+2} + \dot{y}_{i,j+1}) \qquad \text{(C-2)}$$

The use of the above equations requires that four evenly spaced values of the dependent variables be known. These are found in this subroutine by means of a fourth-order Runge-Kutta method (Reference 3). To find the values of the y_i's at the (j+1)th step, the following equation is used:

$$y_{i,j+1} = y_{i,j} + \frac{1}{6} (k \quad + 2k_{i,2} + 2k_{i,3} + k_{i,4}) \qquad \text{(C-3)}$$

where

$$
\begin{aligned}
k_{i,1} &= hf_i \; (t_j,\ x_{1,j},\ x_{2,j},\ldots,\ x_{n,j}) \\
k_{i,2} &= hf_i \; (t_j + \tfrac{1}{2}\, h,\ x_{i,j} + \tfrac{1}{2}\, k_{1,1},\ \ldots,\ x_{n,j} + \tfrac{1}{2}\, k_{n,1}) \\
k_{i,3} &= hf_i \; (t_j + \tfrac{1}{2}\, h,\ x_{1,j} + \tfrac{1}{2}\, k_{1,2},\ \ldots,\ x_{n,j} + \tfrac{1}{2}\, k_{n,2}) \\
k_{i,4} &= hf_i \; (t_j + h,\ x_{1,j} + k_{1,3},\ \ldots,\ x_{n,j} + k_{n,3})
\end{aligned}
\tag{C-4}
$$

Thus, given initial values of the dependent variables, the y_i's, the independent variable, t, and the integration interval size, h, the differential equations are integrated three steps using Equations (C-3) and (C-4). At this point, the integration is continued using Equations (C-1) and (C-2).

A discussion of both the Adams and Runge-Kutta methods is presented in Reference 3. From this reference, the truncation error, Δy, at a given step can be shown to be

$$\Delta y = \left(\frac{y_{i,j+4} - y_{i,j+4}^{(p)}}{14.2}\right) \tag{C-5}$$

so that the absolute error is

$$\Delta y_{ABS} = |\Delta y|$$

and the relative error is

$$\Delta y_{REL} = \frac{\Delta y_{ABS}}{|y_{i,j+4}|} \tag{C-6}$$

At the end of each integration step, error tests could be performed and the integration interval, h, adjusted accordingly. In the present version of the program a fixed interval size is used and no attempt is made to satisfy error specifications.

The quantities in the parameter list are:

H	current value of the integration interval
DS	integration interval
Y	array containing current values of the dependent variables
DY	array containing current values of the derivatives of the dependent variables
NEQ	number of equations being integrated; routine dimensioned for a maximum of 12
NDIFEQ	control index
S	current value of independent variable

Called by:
TRJTRY

C-4 Subroutine BDCOEF

Subroutine BDCOEF organizes the calculation of the influence coefficient matrices for the elliptic store body at field points $(X_{FP}+X_{WLE}, Y_{FP}, Z_{FP})$. The routine is used to compute the influence of the source panels of the elliptic store at u-velocity panel control points. A listing of this routine is presented in Figure C-1(e) of this report.

Routine BDCOEF performs three functions. It first sets aside temporary locations in blank common to hold the control points and the additional arrays for influence coefficients. Its second function is to copy the u-velocity control points into temporary

field point arrays. The field points are in the empennage coordinate system. To account for the offset of the empennage coordinate system, the X_{WLE} of the leading edge of the empennage from the body nose is added to the control points. To prevent numerical inaccuries associated with field points lying inside the body source panels, all body interference panel control points are radially extrapolated to the actual elliptic body surface in ELLCPT. The velocities computed at these field points are then assumed to act at the panel control points. Its third function is to call FLDAIC to generate the influence matrices for the source panels at the above field points.

The descriptions of the parameters in the argument list follow:

XFP,YFP,ZFP	arrays containing the coordinates in the empennage coordinate system at which the influence coefficient matrices are computed
NFLD	number of field points
BETA	Mach number parameter at which panel solutions are computed
XWLE	distance from empennage leading edge to nose of body
RA,RB	vertical and horizontal semi-axes of elliptic section at body interference shell

Subroutine references:

ELLCPT, FLDAIC

Called by:

CRFWBD

C-5 Subroutine BDUVW

Subroutine BDUVW organizes the calculation of the three velocity components from the influence coefficient matrices generated in BDCOEF. It sets aside temporary array space in blank common for the calculations and calls FLDUVW to compute the velocities. A listing of this routine is presented in Figure C-1(f) of this report. The descriptions of the parameters in the argument list follow:

BDU,BDV,BDW	arrays containing the three orthogonal velocity components in the body source panel coordinate system at the field points defined in routine BDCOEF
NFLD	number of field points at which velocities are computed

Subroutine references:
FLDUVW

Called by:
DEMON2

C-6 Subroutine BDYPR

Subroutine BDYPR computes the linear and nonlinear pressures at points on the empennage interference shell. The body is assumed to be covered with interference shell panels and the points coincide with the panel control points. The body meridians on which the points lie pass through the control points of the body interference panels. A listing of this routine is presented in Figure C-1(f) of this report.

BDYPR performs an outer loop on the chordwise number of panels on the interference shell with a second inner loop on the number of panels on a ring to compute the pressure at the panel control points. Routine VELNOR is called to compute the three components of velocity (UCHK,VCHK,WCHK) at the panel control points due to u-velocity singularity strengths. The total velocity, given by Equation (C-7), used to determine the panel pressure is composed of the sum of the velocities induced by u-velocity panels, induced by body source panels, induced by discrete vortices, and contributions due to pitching, yawing, and rolling motion.

$$\begin{aligned} u &= U_{CHK} + U_{BD} + U_{DMP} \\ v &= V_{CHK} + V_{BD} + V_{VRTX} + V_{DMP} \\ w &= W_{CHK} + W_{BD} + W_{VRTX} + W_{DMP} \end{aligned} \tag{C-7}$$

The linear and nonlinear pressures are

$$C_{p_{linear}} = -2 \frac{u}{V_\infty} \tag{C-8}$$

$$C_{p_{nonlinear}} = \frac{2}{\gamma M_\infty^2} \left\{ \left[1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{V_R^2}{V_\infty^2} \right) \right]^{\frac{\gamma}{\gamma - 1}} - 1 \right\} \tag{C-9}$$

where

$$\begin{aligned} \frac{V_R}{V_\infty} &= 1 + \frac{2u}{V_\infty} \cos \alpha_c - \frac{2v}{V_\infty} \sin \alpha_c \sin \phi_r \\ &+ \frac{2w}{V_\infty} \sin \alpha_c \cos \phi_r + \frac{u^2 + v^2 + w^2}{V_\infty^2} \end{aligned} \tag{C-10}$$

The descriptions of the parameters in the argument list follow:

NDAMP option to include p,q,r velocity contributions (0=no, 1=yes)

XM moment center

VSTORE store total velocity

VAR array containing current state of trajectory variables

Subroutine references:
VELNOR, VELDMP

Called by:
SPECPR

C-7 Subroutine BVARIA

Subroutine BVARIA calculates the β's based on nonlinear shock wave shapes of the various fixed aircraft components at a point (X_B, Y_B, Z_B). No value is computed for the separated store. A listing of the routine is presented in Figure C-1(f) of this report.

The β's in this routine are computed for the shocks propagating from the fuselage nose, from the rack, and from the fixed stores. They are computed for either the circular or noncircular fuselage options, and for either or both the circular or elliptic stores. The value of β is set equal to the free stream β_∞ for points ahead of the first influence of the nonlinear shock shape and aft of the linear theory Mach wave propagating

from XSHLDR. Between the first shock influence, designated by the parameter BTNOSE (=$\Delta X/\Delta R$), and the return to β_∞, a linear interpolation from BTNOSE to β_∞ is computed based on the axial location at a constant radial distance from the body centerline. The shock shapes for the circular fuselage, rack, and both store types are assumed to be generated at zero angle of attack and the values used here are corrected to include the effects of this body rotation. The noncircular fuselage shock is already generated including angle of attack effects. This value of β is used to compute the influence at that point for all panels or singularities.

The descriptions of the parameters in the argument list follow:

XB,YB,ZB — coordinates of points at which influence is to be computed in fuselage body system in Figure 3

Subroutine references:
SHKLOC, XVSRT

Called by:
DEMON2, SFORCE, SFORC2, SEMFOR

C-8 Subroutine CEL1

Subroutine CEL1 calculates the complete elliptic integral of the first kind. This subroutine has been taken directly from Reference 4. For a description of the routine that reference should be consulted. A listing of the routine is presented in Figure C-1(h). The comment cards give a brief explanation of the use of the routine.

Description of Parameters:

RES	result value
AK	modulus (input)
IER	resultant error code where
	IER=0, no error
	IER=1, AK not in range -1 to +1

Called by:

SEMPIN

C-9 Subroutine CEL2

Subroutine CEL2 calculates the complete elliptic integral of the second kind. This subroutine has been taken directly from Reference 4. For a description of the routine that reference should be consulted. A listing of the routine is presented in Figure C-1(i). The comment cards give a brief explanation of the use of the routine.

Descriptions of Parameters:

RES	result value
AK	modulus (input)
A	constant term in numerator
B	factor of quadratic term in numerator
IER	resultant error code where
	IER=0, no error
	IER=1, AK not in range -1 to +1

Called by:

SEMPIN

C-10 Subroutine CRFWBD

Subroutine CRFWBD reads and prints the input for the u-velocity panels on the empennage, organizes the fin and body interference panel layout, and computes the corresponding influence coefficient matrix. Both the L*U decomposition of the matrix and the various geometric arrays required for the pressure calculations are saved on TAPE3 when multiple sets of fins are present. A listing of this routine is presented in Figure C-1(i) and a flow chart is given in Figure C-5 of this report.

CRFWBD is set up to read the input for the empennage for the elliptic store and to compute the various parameters which remain constant during the trajectory calculations. The routine can accept input for from one to four arbitrarily oriented fins with or without multiple breaks in leading and trailing edge sweep. If the configuration specified is an interdigitated tail, the fins are oriented in symmetric pairs at the angles PHIDIH, and THETIT.

The routine is organized to read and initialize the geometry and then call LAYOUT to compute the panel corner coordinates for each of the separate fins, and call LAYBIP to compute the panel corner coordinates for the body interference panels. A summary print of these arrays is made. BDCOEF is then called to compute the u,v,w influence coefficient matrices for the effect of the store source panels on the empennage panel control points. These matrices are saved on TAPE10. To compute the influence matrix for the empennage, CRFWBD performs an outer loop on the number of influencing panels and an inner loop on the number of influenced panel control points. VELNOR is called to compute the influence of the panel in the local fin or panel coordinates at the control point.

Routine ROTWB or ROTWF are used to transform the coefficient back to the empennage coordinate system. The FVN array containing the panel coefficients is decomposed by PAS001. If multiple empennages are to be considered all arrays are saved on TAPE3 by a call to EGMSAV. A copy of the influence coefficient matrix, FVN, is also saved on TAPE3 for later use.

The descriptions of the parameters in the argument list follow:

AMACH	free stream Mach number
REFA	reference area, square feet
REFD	reference diameter of body, feet
XLE	x location in store source panel coordinates of leading edge of wing, feet
RAZ	semi-axis in vertical direction of elliptic body, feet
RBY	semi-axis in horizontal direction of elliptic body, feet
KRAD	number of meridional lines used to define panel boundaries about the elliptic body
MLTFIN	logical variable option for multiple fins: TRUE=multiple empennages; FALSE=single empennage
HEAD	array containing alphanumeric heading for empennage

Subroutine references:

BDCOEF, EGMSAV, IOWRIT, LAYBIP, LAYOUT, PAS001, ROTWB, ROTWF

Called by:
    TRJTRY

C-11 Subroutine DASCRU

Subroutine DASCRU performs the numerical integration of the function F for routine VPATHL. The routine varies the step size to obtain best accuracy during the integration of the vortex paths. A listing of the routine is presented in Figure C-1(1) of this report.

The descriptions of the parameters in the argument list follow:

A	initial time or spacial variable
B	final time of spacial variable
H	time or spacial step size
N	number of variables to be integrated
XO	array of dependent variables to be integrated
WK	work array
IER	error return index
E5	1/2 desired resolution

Subroutine references:
    F

Called by:
    VPATHL

C-12 Function DBLU

The complex function DBLU calculates the intermediate transform variable w for the conformal transformation of an elliptical body with wings to the transform (circle) plane. The equation programmed is described in Equation (I112) in Appendix I of Reference 5. This routine is called from the complex function Z.

A listing of this routine is presented in Figure C-1(m) of this report.

The description of the parameter in the argument list follows:

Z complex location of control point in crossflow plane Z=complex (YCP,ZCP)

Called by:

Z

C-13 Subroutine DEMON2

Subroutine DEMON2 calculates the right-hand side for the boundary conditions of the u-velocity panels on a single finned section, calls for the solution, and calls SPECPR for the computation of pressure distributions and loads. The right-hand side is computed including the nonuniform flow field of the parent aircraft, the presence of the store body source panels and their image reflection from the fuselage or wing, the presence of discrete vortices, and the influence of the angular deflection of control surfaces. A listing of the routine is presented in Figure C-1(n) of this report. A flow chart of DEMON2 is presented in Figure C-6. This routine is an adaptation of the methods and equations described in Sections 3.4 and 4.3 of Reference 5 to a nonuniform flow field.

DEMON2 requires the calculation of two sets of matrices on external files prior to its call. The first is the set of u,v,w influence function matrices computed by BDCOEF for the influence of elliptic body source panels at empennage control points. These matrices are saved on file TAPE10. The second matrix, the influence coefficient matrix for the empennage on itself computed in CRFWBD, is saved on TAPE3.

The primary loop in this routine defines the right-hand side, RHS, of the velocity equations for each of the NWBP u-velocity panels. The first NPANLS equations covering the fins are defined in terms of the velocity normal to the surface. The next NBIP panels covering the body interference panels are set equal to zero. All angle of attack and parent aircraft effects are assumed to be included in the source panels spanning the same portion of the body. The velocities summed at each of the NPANLS fin panels used to compute the boundary condition are

$$U_{ext} = -U_{A/C}\left(\frac{V_\infty}{V_{\infty_s}}\right) + \frac{V_x}{V_{\infty_s}} + U_{BD} + U_{DMP}$$

$$V_{ext} = V_{A/C}\left(\frac{V_\infty}{V_{\infty_s}}\right) - \frac{V_y}{V_{\infty_s}} + V_{BD} + V_{VRTX} + V_{IFIN} + V_{DMP}$$

$$W_{ext} = -W_{A/C}\left(\frac{V_\infty}{V_{\infty_s}}\right) + \frac{V_z}{V_{\infty_s}} + W_{BD} + W_{VRTX} + W_{IFIN} + W_{DMP}$$

The parent aircraft velocities, $U_{A/C}$, $V_{A/C}$, $W_{A/C}$, are computed in the parent aircraft system by RESVEL. They are rotated into the store body axes and must be ratioed by V_∞/V_{∞_s} to account for the change in store velocity relative to the free stream. The components V_x, V_y, V_z are the free-stream contribution due to the translational motion of the store as computed in VXYZ. The body source panel induced velocities, U_{BD}, V_{BD}, W_{BD}, on fin u-velocity panels are computed in BDUVW. If vortices are present, their effect is computed by a call to VVELS as V_{VRTX}, W_{VRTX}. These velocities are computed at fin control points and at the projection of body interference panel control points onto an ellipse 1.0 percent larger than the actual elliptic shape. Only the trailing-edge vortices from the first empennage may currently be accounted for. The velocities due to the body and vorticity computed on the interference shell are not used in

the specification of the right-hand side, but are used later in the computation of the pressures. The velocities induced by the image store body, V_{IFIN}, W_{IFIN}, are computed in IMAGFN. And lastly, the contribution of the rotational motion of the store, U_{DMP}, V_{DMP}, W_{DMP}, are computed in VELDMP and added to the totals.

To specify the right-hand side the velocities in the body axes are rotated into the local panel coordinates by ROTWF. The right hand side is thus defined

$$RHS = -W_w - \sin\delta_i \qquad i = R, L, U, D$$

and

$$\begin{pmatrix} V_v \\ W_w \end{pmatrix} = [ROT_{WF}] \begin{pmatrix} V_{ext} \\ W_{ext} \end{pmatrix}$$

where the δ_i are the deflections of the fin surfaces for the i'th fin.

The descriptions of the parameters in the argument list follow:

VAR(N), N=1,2,...12	$\dot{\xi}$, $\dot{\eta}$, $\dot{\zeta}$, p, q, r, ξ, η, ζ, Ψ, Θ, Φ
DC	direction cosine matrix, A, Equation (B-2), Reference 2
XMOM	moment center relative to store nose, feet
VSTORE	V_{∞_s}, Equation (97), Reference 2
NDAMP	damping indicator, see input item 4
FCOEF	array containing empennage force and moments; C_Y, C_N, C_ℓ, C_m, C_n, respectively

XLE — x-station of leading edge of empennage coordinates relative to body nose (positive aft)

HEAD — alphanumeric title of empennage

Subroutine references:

BDUVW, BVARIA, ELLCPT, INTOST, IMAGFN, IOREAD, PAS002, RESVEL, ROTWF, SPECPR, STTOIN, VELDMP, VVELS

Called by:

TRJTRY

C-14 Subroutine DIRCOS

Subroutine DIRCOS computes the direction cosines which relate the store body coordinate system to the inertial coordinate system. The direction cosines are given by Equation (B-2) of Reference 2. A listing of the routine is presented in Figure C-1(o). The three angles Ψ, Θ, and Φ are brought into the subroutine in A(10), A(11), and A(12), respectively, and the direction cosines are returned in the D array.

Called by:

VXYZ

C-15 Complex Function DSDZ

Function DSDZ computes the complex derivative which is used in Equation (I124) of Reference 5 for the velocity at a given point due to vortices and in Equation (I134) for velocities due to pitch, bank and vorticity. The methods and equations used here are described in Equation (I126) in Section 5.3 of Appendix I of Reference 5. A listing of the routine is presented in Figure C-1(o) of this report.

Called by:
    PITROL, VOTEX, VVELS

C-16 Complex Function DZDS

Function DZDS computes the complex derivative which is used in Equation (I124) of Reference 5 for the velocity at a given point due to vortices and in Equation (I134) for velocities due to pitch, bank and vorticity. The methods and equations used here are described in Equation (I125) in Section 5.3 of Appendix I of Reference 5. A listing of the routine is presented in Figure C-1(o) of this report. The description of the argument list follows:

S	complex function of Z(X,Y)

Called by:
    VOTEX

C-17 Complex Function D2SDZ2

Function D2SDZ2 computes the complex second derivative which is used in Equation (I124) of Reference 5 for the velocity at a given point due to vortices and in Equation (I134) for velocities due to pitch, bank and vorticity. The methods and equations used here are described in Equation (I128) in Section 5.3 of Appendix I of Reference 5. A listing of the routine is presented in Figure C-1(p) of this report. The description of the argument list follows:

S	complex function of Z(X,Y)

Called by:
    VOTEX

C-18 Subroutine EGMSAV

Subroutine EGMSAV saves the common blocks generated by routine CRFWBD that are required by DEMON2 to complete the solutions for the elliptic store multiple interference shells. This routine is used to save the empennage data arrays on TAPE3 when two sets of fins and interference shells exist. EGMSAV writes unformatted all the variables in common blocks ONE, THREE, SWEEPS, INTRDT, and WBTR on TAPE3. The arrays in these common blocks constitute the minimum information that may be saved to restart the empennage calculations. A listing of this routine is presented in Figure C-1(p) of this report.

Subroutine references:
IOWRIT

Called by:
CRFWBD

C-19 Subroutine EGMRST

Subroutine EGMRST restores the data in the common blocks saved by routine EGMSAV. The information is read unformatted from TAPE3. Use of TAPE3 allows routines CRFWBD and DEMON2 to be used for more than one empennage. The arrays read define all the variables in common blocks ONE, THREE, SWEEPS, INTRDT, and WBTR. A listing of the routine is presented in Figure C-1(p) of this report.

Subroutine references:
IOREAD

Called by:
TRJTRY

C-20 Subroutine EJECTR

Subroutine EJECTR calculates ejector forces and moments acting on the separating store. It is called from three different places in TRJTRY. A listing of the routine is presented in Figure C-1(p) of this report.

The first call to EJECTR with the variable NJECTR=1 causes the portion of the routine which reads and prints the ejector input data to be executed. The input data are items 5 through 8 to Program II and are described in Section 4.2.2 of Volume II (Reference 1). Before returning to TRJTRY, NJECTR is set equal to 2.

The second call to the routine with NJECTR=2 causes transfer to the part of the routine which locates the ejector feet, one or two, when the separating store is in the carriage position. The locations of the feet are calculated in the inertial coordinate system and stored in the XYZEI array. Upon returning to TRJTRY, NJECTR is set equal to 3.

The third call to EJECTR occurs within the trajectory integration loop of TRJTRY with NJECTR=3. The first steps in this section of the routine are to locate the store moment center and one other point on the store axis in the inertial coordinate system. The ejector forces and moments are then initialized to zero. The remainder of the routine is a DO loop over the number of feet. The store radius at the axial location of the foot is determined and then calculations are performed which determine the Y,Z coordinates, in the inertial system, of the point at which the foot strikes the store body. These coordinates are used along with the foot location in the carriage position, previously calculated, to determine the distance the foot has traveled. If the distance exceeds the input stroke length, forces and moments for this foot

are not calculated. If the foot travel does not exceed the stroke length, the total ejector force is calculated and resolved into components in the inertial system.

The last parts of this loop calculate the forces and moments due to the ejector foot in the store coordinate system. The first step is to take the forces in the inertial coordinate system and using subroutine INTOST project them into the store system. The point at which these forces act is found by, first, locating the point in the inertial system relative to the store moment center and then, with a call to INTOST, locating the point in the store system. With this point determined, the ejector foot induced moments are calculated.

The descriptions of the parameters in the subroutine argument list follow:

DC	direction cosines relating the inertial coordinate system to the store coordinate system
VAR	array containing the dependent variables in the trajectory integration; $\dot{\xi}$, $\dot{\eta}$, $\dot{\zeta}$, p, q, r, ξ, η, ζ, Ψ, Θ, Φ
DTOR	degrees to radians conversion factor; $\pi/180^{\circ}$
TIME	current value of time in the trajectory calculation measured from the beginning of the trajectory, sec.
STLGTH	length of separating store, feet
XMOM	store moment center location measured relative to store nose, positive behind nose; feet

NPOLY number of polynomials describing circular store body shape

COEF array containing the coefficients of the NPOLY polynomials

XEND array containing values of x/ℓ of the end points of the NPOLY polynomials

ELLSTR store type logical indicator;
ELLSTR=true, elliptic store
ELLSTR=false, circular store

Subroutine references:
SHAPE, STTOIN, INTOST

Called by:
TRJTRY

C-21 Subroutine ELI1

Subroutine ELI1 calculates the general elliptic integral of the first kind. This subroutine has been taken directly from Reference 4. For a description of the routine that reference should be consulted. A listing of the routine is presented in Figure C-1(r). The comment cards give a brief explanation of the use of the routine.

Description of Parameters:

RES result value

X upper integration bound (argument of elliptic integral of first kind)

CK complementary modulus

Called by:
SEMPIN

C-22 Subroutine ELI2

Subroutine ELI2 calculates the general elliptic integral of the second kind. This subroutine has been taken directly from Reference 4. For a description of the routine that reference should be consulted. A listing of the routine is presented in Figure C-1(s). The comment cards give a brief explanation of the use of the routine.

Description of Parameters:

R result value

X upper integration bound (argument of elliptic integral of second kind)

CK complementary modulus

A constant term in numerator

B quadratic term in numerator

Called by:
SEMPIN

C-23 Subroutine ELLCPT

Subroutine ELLCPT performs a radial extrapolation of the Y,Z coordinates of a control point of a body interference panel on an elliptic store to a point on the surface of the ellipse. This routine compares the radial distance from the centerline to the control point, RCPT, to the radial distance to the surface of the ellipse, RBODY. If the control point lies within the ellipse, the control points are redefined to lie on the surface using the following equations:

$$R_{CPT} = \sqrt{Y_{CP}^2 + Z_{CP}^2}$$

$$R_{BODY} = \frac{1.0}{\sqrt{\left(\frac{Z_{CP}}{R_{CPT}\;R_A}\right)^2 + \left(\frac{Y_{CP}}{R_{CPT}\;R_B}\right)^2}}$$

$$Y_{CP} = \left(\frac{R_{BODY}}{R_{CPT}}\right) Y_{CP_{old}}$$

$$Z_{CP} = \left(\frac{R_{BODY}}{R_{CPT}}\right) Z_{CP_{old}}$$

A listing of this routine is presented in Figure C-1(t) of this report. The descriptions of the parameters in the argument list follow:

YCP,ZCP	Y and Z ordinates of control point on surface of panel

RA Semi-axis of ellipse in vertical direction

RB Semi-axis of ellipse in horizontal direction

Called by:
BDCOEF, DEMON2

C-24 Subroutine ELLSHP

Subroutine ELLSHP computes the horizontal and vertical semi-axes and their axial derivatives for an elliptic store shape. The calculation is performed by table look up in the elliptic store input geometric shape. For a given axial station the closest two input sections are located and the axes and their derivatives computed by linear interpolation between them. The parameter IGROW is also set to one if the derivative of either of the axes is greater than 0.002 to indicate an expanding or contracting cross section is to be used in VPATHL. A listing of the routine is presented in Figure C-1(t) of this report. The descriptions of the parameters in the argument list follow:

X axial station (positive aft) at which geometry is computed

BY,AZ horizontal and vertical semi-axes of ellipse at X

BYP,AZP dBY/dx and dAZ/dx, derivatives at X

Called by:
F, VPATHL

C-25 Subroutine ELRFLB

Subroutine ELRFLB computes the parameters which locate the image elliptic store relative to the fuselage. The fuselage may

either be circular or noncircular. The basic calculations performed here are in the fuselage coordinate system of Figure 5 Volume II, although four different coordinate systems are involved. A listing of the routine is presented in Figure C-1(t) of this report.

The sequence of calculations for the locations of the image store and the first influence of the reflected shock on the real stor are as follows. ELRFLB first computes the location of the store nose in fuselage coordinates (XBN,YBN,ZBN), the meridian angle between the store and fuselage centerlines ($\phi_{SF} = \tan^{-1}(-YBN/-ZBN)$), and the angle of the fuselage with the store axis, $\phi_{SHK} = \phi_{SF} + \Phi$. A table of r_{SHK} versus x_{SHK} relative to the store is then generated in the plane containing the store nose and the fuselage centerline at ϕ_{SHK}, and rotated into the fuselage body axes of Figure 5 of Volume II. The table is generated by XVSRT from the elliptic store shock shape at zero degrees angle of attack rotated to $\alpha(1-\varepsilon_{\alpha})$. The location of the intersection of the above tabulated shock shape with the fuselage at X_{BS}, R_{BS} relative to the store nose is computed in SWINT when the circular store model is used and in SWINTE when the noncircular store model is present. The location of the store and the store shock reflecting from the fuselage are shown pictorially in Figure C-7.

The angle the shock reflects from the fuselage surface is computed from the angle the shock strikes the fuselage, θ_s, plus twice the angle of the local fuselage surface slope, θ_t.

$$\theta_r = \theta_s + 2\theta_t$$

The intersection of the reflected shock with the store body is computed from the intersection of the line through the store centerline and the straight line projection of the shock at the angle, θ_r, from the point of contact on the fuselage.

Equation of store axis

$$R_{BR} = R_{BN} + \tan\alpha \cdot (X_{BR} - X_{BN})$$

Equation of reflected shock

$$R_{BR} = R_{BS} + \tan\theta_r \cdot (X_{BR} - X_{BS})$$

where

$$\tan\alpha = \frac{R_{CG} - R_{BN}}{X_{BCG} - X_{BN}}$$

The point at which the reflected shock intersects the store axis is

$$X_{BR} = \frac{R_{BS} - R_{BN} + X_{BN} \cdot \tan\alpha - X_{BS}\tan\theta_r}{(\tan\alpha - \tan\theta_r)}$$

$$R_{BR} = R_{BN} + \tan\alpha \cdot (X_{BR} - X_{BN})$$

Tests are made to see whether the two lines are parallel, whether the reflected angle is close to 90°, and whether the intersection point on the store axis is aft of the last store station. The error indicator IMFSTR is set equal to zero for the first and last case. If the angle $\theta_r = 90°$, X_{BR} is set to X_{BS}. The last parameters computed define the location and β's associated with the image store. Their description and the descriptions of the parameters in the argument list follow. Their pictorial representations are found in Figure C-7.

ALPHAC — store included angle of attack at local flow condition

PHIR — store roll orientation relative to flow angle of attack

VAR12	store inertial roll orientation; see VAR(12) in TRJTRY
RSSC	distance from store nose to fuselage centerline; locates line source image on fuselage centerline
RSS	location of line source image and doublet image; $RSS = RSSC-RBS^2/RSSC$
BTNOSC	centerline line source image nose BETA (=XSR/RSSC)
BTNOSF	doublet and line source pair image nose BETA (=XSR/RSS)
SFAC	image line doublet multiplying factor ($=RBS^2/RSSC^2$)
XBR,RBR	axial and radial distance from fuselage in plane of reflection to intersection with store centerline
XSR	axial distance from store nose to point of contact of shock reflection
BTNOSN	image store nose BETA (=XSR/2(RBN-RBS)) computed for shock reflected from fuselage striking store centerline at XSR

Subroutine references:
STTOIN, SWINT, SWINTE, XVSRT

Called by:
SFORC2

C-26 Subroutine ELRFLW

Subroutine ELRFLW calculates the shock shape between the store and the wing surface and calls REFSHK to compute the location of the wing image store. The angle, PHI, is computed to define the location of the plane of the shock shape relative to the store axes to be used for the reflection calculation. A table of shock X and R values are generated by successive calls to XVSRT. Routine REFSHK uses this tabulated shape to find the wing reflection intersection and image store location. A listing of the routine is presented in Figure C-1(u) of this report. The descriptions of the parameters in the argument list follow:

ALPHAC	store included angle of attack at local flow condition
PHIR	store roll orientation relative to flow angle of attack
VAR12	store inertial roll orientation; see VAR(12) in TRJTRY

Subroutine references:
REFSHK, XVSRT

Called by:
SFORC2

C-27 Subroutine EXPAND

Subroutine EXPAND calculates the crossflow velocities induced by an expanding or contracting elliptical cross section. The ellipse is assumed to have independently varying elliptic axes,

AZ and BY. The component velocities in the crossflow plane are computed by conformal mapping of the elliptic shape into a circle in the complex plane.

The equations programmed are essentially those in Reference 5, Section 8 of Appendix I, modified for independently varying AZ and BY. A listing of this routine is presented in Figure C-1(u) of this report. The descriptions of the parameters in the argument list follow:

BYP	dBY/dx, rate of change of horizontal elliptic axis with axial distance; positive expanding aft
AZP	dAZ/dx, rate of change of vertical elliptic axis with axial distance; positive expanding aft
X,Y	horizontal and vertical location in crossflow plane of points at which velocities are computed
VS,WS	horizontal and vertical components of velocity in crossflow due to unit velocity normal to section

Called by:
F

C-28 Subroutine F

Subroutine F is called in the vortex path integration scheme to compute the total local crossflow velocity in the vicinity of the body alone due to pitch, sideslip, body volume, and the presence of other vortices. This routine is adapted from Appendix L of Reference 5 and is called from DASCRU to compute the velocities required for the path integration. The crossflow velocities due to pitch and angle or sideslip are computed in

PITROL. The components due to body volume are computed in EXPAND and the presence of other vortices in VOTEX. The resulting components are summed, normalized and saved in the temporary work vector, WK. A listing of this routine is presented in Figure C-1(u) of this report.

The descriptions of the parameters in the argument list follow:

XO	vector containing Y and Z-coordinates of points at which velocities are to be computed
PX	axial station of section under consideration
N	number of independent variable; unused
WK	work array containing normalized velocities on return

Subroutine references:
ELLSHP, EXPAND, PITROL, VOTEX

Called by:
DASCRU

C-29 Subroutine FLDAC2

Subroutine FLDAC2 specifies the local panel geometry for the generation of the influence coefficients of the source panels in a single segment at a prescribed set of field points. Coefficients for each of the orthogonal velocity components are computed for a ring on each of the field points and saved on an external file. This routine is called by FLDAIC. A listing of this routine is presented in Figure C-1(v) of this report.

Routine FLDAC2 performs the analogous function as BODVEL, except that it computes influence coefficients for panels at field points. The routine performs the outside loop on the number of rings in the segment. The middle loop initializes the panel geometry and control points in the local panel coordinates. The innermost loop calls PANVEL to calculate the influence of the panel at each of the field points. At the end of the middle loop the coefficient array for the influence of the panels in the ring under consideration on all the field points is written unformatted onto external file, TAPE10. Additional logic is also included to skip unnecessary calculations. If the sum of the squares of each of the component coefficients is zero for an entire ring of panels at a single field point, all subsequent rings downstream in supersonic flow are set to zero also.

The descriptions of the parameters in the argument list follow:

XPT,YPT,ZPT	arrays of the coordinates of the source panel control points for the entire body
THET	array of inclination angles for panels
DELTA	array of incidence angles for panels
XFP,YFP,ZFP	arrays of coordinates of field points
UB	temporary array dimensioned UB(3,NFLD,JR) to contain u,v,w influence coefficients for a ring on all field points
SVN	temporary array used to test influence of ring on point
LSKP	temporary array used as logical indicator of no further influence at point

NFLD number of field points

XC,YC,ZC arrays of coordinates of panel corners for segment

KFUSOR number of axial station bounding panels in segment

IXZSYM body symmetry indicator

IFU segment number index

JR number of panels in ring J

Subroutine references:
IOWRIT, PANVEL

Called by:
FLDAIC

C-30 Subroutine FLDAIC

Subroutine FLDAIC organizes the generation of the influence coefficients due to body source panels at field points, XFP, YFP, and ZFP. This routine performs the loop on the number of body segments. For each body segment the starting locations in blank common of the panel geometry arrays are initialized and FLDAC2 called to compute the influence coefficients. A listing of this routine is presented in Figure C-1(v) of this report.

The descriptions of the parameters in the argument list follow:

XFP,YFP,ZFP arrays of coordinates of field points

SVN	temporary array used to test influence of ring on point
LSKP	temporary array used as logical indicator of no further influence at point
UB	temporary array dimensioned UB(3,NFLD,JR) to contain u,v,w influence coefficients for a ring on all field points. JR = number of panels in ring
NFLD	number of field points

Subroutine references:
FLDAC2

Called by:
BDCOEF

C-31 Subroutine FLDUVW

Subroutine FLDUVW computes the U,V,W velocity components at field points from the influence coefficient matrices stored on external file TAPE10. The routine performs the matrix multiplication of the coefficients times the panel strengths to get the velocities as follows.

$$U = [UB_1] \cdot GB$$
$$V = [UB_2] \cdot GB$$
$$W = [UB_3] \cdot GB$$

The coefficient matrices are stored in blocks corresponding to the influence of a ring of panels on all the field points. A listing of the routine is presented in Figure C-1(w) of this report. The descriptions of the parameters in the argument list follow:

UB	temporary array of size 3*NFLD*JR to hold influence coefficients. JR = number of panels in ring
GB	arrays of panel strengths
U,V,W	computed orthogonal velocity components in source panel coordinate system at field points
NFLD	number of field points
JROW(J)	array of NRING values of number of panels in Jth ring
NRING	number of ring of panels

Subroutine references:
IOREAD

Called by:
DEMON2

C-32 Subroutine FLDVEL

Subroutine FLDVEL is used to organize the computation of the u, v, w velocity components at the field points XFP, YFP, ZFP. This routine initializes all velocities to zero and computes the locations of the required geometric and strength arrays in blank common for the fuselage or store at hand. If the configuration models the fuselage with inlets, variable INLET is initialized to TRUE to indicate presence of inlet panels. It then performs the looping for the number of segments and the number of rings in each segment to sum the component velocity contributions at each field point. The variable IDLAST is incorporated in this version to permit skipping program initialization logic when making multiple

calls with the same configuration data. A listing of the routine is presented in Figure C-1(w) of this report.

The descriptions of the parameters in the argument list follow:

XFP,YFP,ZFP	arrays of coordinates of field points in coordinate system of body panels at which component velocities are computed
SVN	temporary array of length NFLD used to test for zero influence at field point
LSKP	temporary logical array of length NFLD used to test for zero influence of ring of panels at field point
U,V,W	arrays of component velocities computed for influence of body at field points
NFLD	number of field points
ID	array containing the same variables in the same order as in common DIMENS
IG	index specifying IGth strength solution, GB, to be used

Subroutine references:
FLDVL2

Called by:
IMAGFN, RESVEL

C-33 Subroutine FLDVL2

Subroutine FLDVL2 computes the three components of velocity induced at specified field points by the source panels of a given ring of body panels. The field point is assumed to have no incidence or inclination relative to the flow. A listing of the routine is presented in Figure C-1(x) of this report.

The routine first initializes the field point inclination and incidence transformations to zero. For each of the influencing panels on a ring the control point coordinates, inclination and incidence angles and corner points in the local panel system are defined. For each panel, β_ℓ is initialized to β_o. For panels used to model inlet openings, β_ℓ is set to the minimum of β_o and BTINLT. Further, if an inlet panel slope exceeds BTINLT, β_ℓ is set to $0.99/\tan\delta$. For each of the field points the influence of the panel on the field point is computed, multiplied times the panel strength and summed.

To minimize computational time FLDVL2 tests whether each ring of panels contributes to the total velocity at a given field point. For each point the following sum is made for each panel on the ring

$$SVN(I) = UB^2 + VB^2 + WB^2$$

where UB, VB, WB are the influence coefficients of a panel at the Ith field point. If the net influence for a complete ring, SVN is equal to zero the logical variable LSKP(I) for that field point is set .TRUE. and all further calculations at that point are suppressed. This is based on the assumption that at supersonic speeds once a point is ahead of the Mach waves from a ring, the influence of that ring and all subsequent rings will vanish.

The descriptions of the parameters in the argument list follow:

XPT,YPT,ZPT	arrays of the coordinates of the source panel control points for the entire body
THET	array of inclination angles for panels
DELTA	array of incidence angles for panels
GB	array of panel strengths
XFP,YFP,ZFP	arrays of coordinates of field points
U,V,W	arrays of orthogonal velocity components at panels in x, y, and z directions

SVN	temporary array used to test influence of ring on point
LSKP	temporary array used as logical indicator of no further influence at point
NFLD	number of field points
XC,YC,ZC	arrays of coordinates of panel corners for segment
KFUSOR	number of axial station bounding panels in segment
IXZSYM	body symmetry indicator
JR	number of panels in ring J
JG	starting offset index of first panel in ring
L	index of trailing x-station in segment

Subroutine references:

PANVEL, INLTST

Called by:

FLDVEL, IMAGEV, IMAGFN

C-34 Subroutine FORMOM

Subroutine FORMOM calculates the forces and moments on a non-circular body. It sums the three components of force and three moments calculated from the pressures predicted on the surface of the source panels in an arbitrary flow field. A sum of the forces and moment contributions due to each ring of panels are also computed. A listing of the routine is presented in Figure C-1(y) of this report.

FORMOM performs three DO loops in summing the forces and moments. The outermost loop steps through the number of body segments which in the present use is restricted to one. The middle loop steps through the number of rings in each segment. The contribution of each ring to the normal

and side forces and the three moments are also saved. The contribution of each individual panel is computed in the innermost loop over the number of panels in each ring. The force contributions of each panel are computed as

$$C_{NORM} = CP \cdot A$$

$$\Delta C_{XB} = -C_{NORM} \sin \delta$$

$$\Delta C_{YB} = -C_{NORM} \cos \delta \sin \theta$$

$$\Delta C_{ZB} = C_{NORM} \cos \delta \cos \theta$$

$$\Delta C_{MB} = -\Delta C_{ZB}(X_P - X_{REF}) + \Delta C_{XB}(Z_P - Z_{REF})$$

$$\Delta C_{NYAW} = -\Delta C_{YB}(X_P - X_{REF}) + \Delta C_{XB}\, Y_P$$

$$\Delta C_{LROL} = -\Delta C_{ZB}\, Y_P + \Delta C_{YB}(Z_P - Z_{REF})$$

where δ is the panel incidence angle and θ is the panel inclination angle.

The resulting forces and moments have the following sign conventions. C_{NORM} is the force normal to the panel, positive outward. C_{XB} is positive aft. C_{YB} is positive to the right. C_{ZB} is positive up. C_{MB} is the moment, positive nose up. C_{NYAW} is the moment, positive nose right viewing forward. C_{LROL} is the moment, positive clockwise viewed forward. The above coefficients are dimensional as written. At the end of the three DO loops, the forces are nondimensionalized by REFA and the moments by REFA*REFD.

This routine is written to handle both symmetric and nonsymmetric geometries. The separating store is paneled on both halves so that the nonsymmetric option is always used. Even though the store is symmetric geometrically, the nonuniform velocity field produces a nonsymmetric force distribution. The descriptions of the parameters in the argument list follow:

XPT,YPT,ZPT	arrays of coordinates of control points of panels at which forces and moments are assumed to act
THET	array of inclination angles of panels, radians
DELTA	array of incidence angles of panels, radians
AREA	array of panel areas
CP	array of pressures acting on panel
XC	array of x-values of leading and trailing edges of panels

Called by:
SDSTN2

C-35 Subroutine FREFSH

Subroutine FREFSH calculates the location at which the circular store nose shock reflected by either the circular or noncircular fuselage strikes the store axis and the value of BETA to be used for the image store. This routine is used to locate the image store relative to either the line source model or source panel model of the fuselage. A listing of the routine is presented in Figure C-1(y) of this report. Except where indicated the coordinates used in this routine are in the store body system (Figure 17 of Volume II). A detailed explanation of the equations used in calculating the image store effects are given in Section 4.4 of Reference 2. Some of the following quantities are shown pictorially in Figure C-7.

The procedure used in this routine to generate the parameters necessary to locate and compute the influence of the image circular store is developed as follows.

(1) Compute the location (X_{BN}, Y_{BN}, Z_{BN}) and radial distance from store nose to fuselage axis.

(2) Using the table of R vs X in the circular store coordinates, locate the intersection with either the circular or non-circular fuselage surface (X_{BS}, R_{BS}) in the plane between the store nose and the fuselage axis and the reflection angle $\theta_r = \tan^{-1}(DR/DX)_S + 2$[illegible]$_t$.

(3) Locate store center of gravity $(X_{BCG}, Y_{BCG}, Z_{BCG})$ and define line through store nose and C.G.

(4) If reflection angle exceeds store angle of attack, reflected shock is assumed to miss store.

(5) If $89.5 < \theta_r < 90.5$, intersection with store is set to shock impact point on fuselage.

(6) Otherwise calculate intersection point X_{BR}, R_{BR} with store axis (note: $\alpha_f = -\alpha$ in Figure C-7).

$$X_{BR} = \frac{(R_{BN} - R_{BS} + X_{BN} \tan \alpha_f - X_{BS} (dR/dX)_r)}{(\tan \alpha_f - (dR/dX)_r)}$$

$$R_{BR} = R_{BN} + \tan \alpha_f \, (X_{BN} - X_{BR})$$

(7) Compute distance from store nose to intersection with reflection along store axis, X_{SR}.

(8) If X_{SR} is less than body length, set indicator IMFSTR=1, and define centers of image doublet and two sources and [illegible]'s associated with each.

The descriptions of the parameters in the argument list follow.

NS	number of R versus X values in shock shape
SX,SR	table of X and R defining shape of shock between store nose and fuselage axis

Subroutine references:
SWINT, SWINTE

Called by:
SFORCE

C-36 Subroutine FRSTRT

Subroutine FRSTRT is used to save or restore required program information for the source paneling method to restart a configuration analysis. The routine is set up to allow storage of the information from more than one configuration on the same file. A listing of the routine is presented in Figure C-1(z) of this report. Six options are available based on the information to be stored or retrieved and where the information is to reside. They and the functions they perform are:

KODE	DESCRIPTION
1	saves control integers and arrays in blank common
2	restores control integers and arrays in blank common
3	saves above information and AIC, U, V, and W velocity matrices
4	restores above information and AIC, U, V, and W velocity matrices
5	restores control integers and arrays stored under either option 1 or 3 into starting core location in blank common. All arrays are copied

into blank common rather than into original labeled commons

6 restores information in manner of KODE=5 and reads past velocity matrices stored under option 3 to position file at next record

The information saved under KODE=1 consists of all the variables in labeled commons DIMENS, PARAM, BOPTNS, BGEOM, HEAD, and BSHOCK, and the first NTAP7 variables in blank common. The later set are also saved on TAPE7. If the configuration contains an inlet, in addition the variables in common BINLET and BINSHK are saved. In copying the velocity matrices under KODE=2, the arrays are first copied from TAPE8 or TAPE9 into temporary arrays in blank common and then onto the output file IO. The last two options are used in Program II to stack multiple configurations end-to-end in blank common. Provision has been made for only one configuration with an inlet. Variables from commons BINLET and BINSHK are copied back into those arrays under all KODE options. During this data retrieval three indices used to locate the control variable arrays are saved.

IDO (=LASTA+1) first location in blank common of variables contained in labeled commons DIMENS, PARAM, BOPTNS, BGEOM, and TITLE in order

ISKO (=IDO+571) first location in blank common of variables contained in labeled common BSHOCK

IAO (=ISKO+240) first location in blank common of geometric and strength arrays previously saved on TAPE7

The descriptions of the parameters in the argument list are:

IO external file unit number onto which the data is stored or retrieved in unformatted form

KODE — optional index selecting information to be read or written to file IO; see above descriptions

LASTA — last location in blank common currently defined

Subroutine references:
IOREAD, IOWRIT

Called by:
RDFILE, STRDAT

C-37 Subroutine FXBOD

Subroutine FXBOD sums the source panel force and moment distributions in the axial direction for each of the body sections on an elliptic store. Store section boundaries must correspond to panel edges for proper definition of forces. A listing of the routine is presented in Figure C-1(aa) of this report. The descriptions of the parameters in the argument list follow:

DCYX — array of dC_Y/dx at center of lift of Ith ring of source panels

DCZX — array of dC_N/dx at center of lift of Ith ring of source panels

FBOD(I,J) — array containing Ith net force or moment component for Jth body section

I	1	2	3	4	5
Net load	C_Y	C_N	C_ℓ	C_m	C_n

IBOD array containing index x station of trailing edge of last ring number in Jth body section

NBOD number of body sections

Called by:
TRJTRY

C-38 Subroutine IMAGEV

Subroutine IMAGEV computes the influence of the JBth ring of source panels on the image store on the real store. The resulting velocities normal to the real store panels are computed and subtracted from the boundary conditions of each panel. A listing of the routine is presented in Figure C-1(bb) of this report.

The procedure used in this routine follows. IMAGEV first initializes the segment and ring indices and sets the velocity contribution of the ring to zero. The contribution of the JBth ring on the real store at control point locations on the image store previously generated in IMAGYZ are computed by FLDVL2. The velocities computed for the real store at the image store control points are then transformed to velocities from the image store at the real store control points. The velocities are related by $(v,w)_{\text{image on real}} = (v,-w)_{\text{real on image}}$. This transformation is combined with the resolution of the image store induced velocities in the image system into the real store coordinate system. The combined transformation is

$$\begin{Bmatrix} v \\ w \end{Bmatrix}_{\text{real}} = \begin{bmatrix} \cos 2\phi_I & -\sin 2\phi_I \\ -\sin 2\phi_I & -\cos 2\phi_I \end{bmatrix} \begin{Bmatrix} v \\ w \end{Bmatrix}_{\text{real on image}}$$

where ϕ_I is the angle between the real store z axis and the plane containing the image and real store longitudinal axes. The contribution to the boundary condition is modified using the normal component due to the image store, V_{N_i}, at the influenced panel control point. For the Ith panel

$$V_i = V_i - V_{N_i}$$

where V_i is the total normal velocity at the control point.

The descriptions of the parameters in the argument list follow:

GB array of panel strengths

VB array of panel normal velocities

IMAGE logical indicator of influence of image store ring on real store
 IMAGE=FALSE, no influence felt on real store
 IMAGE=TRUE, ring has influenced real store

JG offset location of end of data for ring of panels last used

JR number of panels in ring of influencing panels

Subroutine references:
 FLDVL2, VNORM

Called by:
 SMARCH

C-39 Subroutine IMAGFN

Subroutine IMAGFN computes the influence of the image store body on fin control points. The velocities are actually computed for the real store at image store locations and transformed back onto real store fins. A listing of the routine is presented in Figure C-1(bb) of this report.

The routine first computes the fin control point on the image store relative to the real store elliptic source panel coordinates by the following transformations.

$$Y_I = \cos \phi_I \, Y_{CPT} + \sin \phi_I \, Z_{CPT}$$

$$Z_I = \sin \phi_I \, Y_{CPT} - \cos \phi_I \, Z_{CPT} + Z_{IP}$$

and

$$Y_{IM} = \cos \phi_I \, Y_I - \sin \phi_I \, Z_I$$

$$Z_{IM} = \sin \phi_I \, Y_I + \cos \phi_I \, Z_I$$

The angle ϕ_I is the angle between the real store z axis and the plane containing the image and real store longitudinal axes ($\phi_I = \phi_{IM} - \Phi$ in Section C-40). The quantity Z_{IP} is the distance between the real store nose and the image store nose.

The slope of the line from the control point to the real store nose, $\beta_{CPT} = \sqrt{Y_{IM}^2 + Z_{IM}^2}/(X_{CPT} + X_{WLE})$, is compared with the β_N associated with the first influence of the image store on the real store. If $\beta_{CPT} < \beta_N$, the point does not feel the image store and the routine returns. If the image store is felt, FLDVEL is called

to compute the source panel influence of the body on the fin control point. The velocities are then transformed back onto the real store by:

$$U_{IFIN} = UP$$

$$V_{IFIN} = \cos 2\phi_I \; V_P - \sin 2\phi_I \; W_P$$

$$W_{IFIN} = -\sin 2\phi_I \; V_P - \cos 2\phi_I \; W_P$$

The descriptions of the parameters in the argument list follow:

XCPT,YCPT, ZCPT	coordinates of fin control point in empennage reference axes
UIFIN,VIFIN, WIFIN	U,V,W component velocities at fin control point induced by image store body in empennage reference axes
XWLE	x-station of leading edge of empennage reference axes in body coordinates

Subroutine references:
FLDVEL

Called by:
DEMON2

C-40 Subroutine IMAGYZ

Subroutine IMAGYZ generates a set of control points at the image elliptic store location for use in computing the influence

of the image store source panels on the real store. The image store control points are computed in the local coordinate system attached to the rotating and translating store. Velocities computed later are transformed to produce the influence of the image store source panels on the real store. The transformation used to locate the image control points are:

Angle between vertical plane and line between noses of real store and image store

$$\phi_{IM} = \tan^{-1} \frac{(Y_{IMO} - Y_{PTO})}{-(Z_{IMO} - Z_{PTO})}$$

Distance between noses of image and real store

$$Z_{IP} = \sin \phi_{IM}(Y_{IMO} - Y_{PTO}) - \cos \phi_{IM}(Z_{IMO} - Z_{PTO})$$

Location of image of Y_{PT}, Z_{PT} on image store in a coordinate system with origin at real store nose and with the z axis lying on the line connecting the store noses.

$$y_i = \cos(\phi_{IM} - \phi) Y_{PT_i} - \sin(\phi_{IM} - \phi) Z_{PT_i}$$

$$z_i = -\sin(\phi_{IM} - \phi) Y_{PT_i} - \cos(\phi_{IM} - \phi) Z_{PT_i} + Z_{IP}$$

Image store point in real store coordinates

$$Y_{IM_i} = \cos(\phi_{IM} - \Phi)y_i + \sin(\phi_{IM} - \Phi)z_i$$

$$Z_{IM_i} = -\sin(\phi_{IM} - \Phi)y_i + \cos(\phi_{IM} - \Phi)z_i$$

A listing of this routine is presented *in Figure* C-1(cc) of this report. The descriptions of the parameters in the argument list follow:

YPT,ZPT	array of Y and Z's of real store control points
YIM,ZIM	array of Y and Z's computed for location of image store relative to real store axes
NBODY	number of control points
VAR12	Φ(=VAR(12)), roll orientation of ejected store
YIMO,ZIMO	coordinates of nose of image store in fuselage axis system
YPTO,ZPTO	coordinates of nose of real store in fuselage axis system

Called by:

SFORC2

C-41 Subroutine IMSVEL

Subroutine IMSVEL calculates the velocities induced by a circular image store at a control point on the real circular store. A listing of the subroutine is presented in Figure C-1(cc). The equations programmed in this routine are derived in Appendix A of Reference 2.

The routine consists of two DO loops followed by a summing up of the axial, radial, and tangential velocity component and a resolution of the latter two into the real store coordinate system.

The first loop calculates the line source induced velocities using Equation (A-15) of Reference 2. The first source has its origin at the image store nose and successive sources have their origins downstream. A test in the loop is made to determine whether a source influences the field point. If it does not, a transfer out of loop takes place since the following sources also cannot influence the field point.

The second DO loop calculates the velocities induced by the two sets of line doublets using Equations (A-24) and (A-32) of Reference 2. The calculation is performed in a manner identical to that previously described for the sources.

Following the second loop the axial, radial, and tangential velocities are summed up and the latter two transformed to v and w velocities in the store coordinate system.

$$v_{SD} = -V_{RAD}\sin\theta + V_{TAN}\cos\theta$$

$$w_{SD} = -V_{RAD}\cos\theta - V_{TAN}\sin\theta$$

The descriptions of the parameters in the subroutine argument list follow:

NM	number of line sources or line doublets
XFP	axial location of field point relative to image store nose
RFP	radial location of field point relative to image store longitudinal axis
CTHETA, STHETA	$\cos\theta$ and $\sin\theta$ where θ is the angle between the image store z axis and the line connecting the store axis with the field point
BETAAL	local value of β used in the line source calculation
BETASL	local value of β used in the line doublet calculation
USD,VSD,WSD	u, v, and w velocity components in the real store coordinate system induced at a control point by the image store
FAC	image store doublet multiplicative factor: FAC = 1.0 for wing image $0 \leq$ FAC ≤ 1.0 for circular fuselage image store, calculated in FREFSH FAC = 0 for circular fuselage centerline image store

Subroutine references:

SDTRMS

Called by:

SDSTRN

C-42 Subroutine INLBET

Subroutine INLBET is used to compute the location of the first influence of the inlet shock at the lateral location Y,Z. The inlet shock axial location is computed by table look-up in the radial traverses generated in Program I. The interpolation procedure follows the sequence outlined in Section A-20 of Volume III. A listing of the routine is presented in Figure C-1(cc).

In computing the shock location relative to the inlet, INLBET makes four assumptions. First, below the inlet, all traverses are computed along constant Y-values (ϕ=0) as shown in the sketch in Section A-20 of Volume III. All interpolations are made linearly between values of Y. Second, traverses are computed along constant ϕ lines originating from the inlet outboard leading-edge XINLT,YINLT,ZINLT. All interpolations are made at a given value of R linearly in angle ϕ. Third, lateral coordinates above the inlet use only the linear theory free-stream value of β_∞ to compute the shock location. Fourth, all calculations are symmetric about the fuselage centerline.

Three parameters are computed, the radial and axial locations, RS and XS, and β_{IS}. The procedure used to compute the inlet shock parameters is as follows. The Y,Z point is first located on the positive side of the symmetric configuration and then in the appropriate region as shown in the sketch in Section A-20. In the first region below the inlet, traverses are computed

in a vertical plane. If only one radial traverse has been computed, the shock shape is assumed constant in the region under the fuselage and inlet. If more than one radial traverse has been generated to define the shock shape under the inlet, the axial shock location is computed at the distance $RS=|Z-ZINLT|$ below the inlet along adjacent traverses in XVSR. The local value of XS is then computed by linear interpolation between values of Y. If the Y-value is less than YCPI, the inner edge of the inlet, the radial shock distance, RS, is computed to reflect the first influence of an inlet panel.

$$RS = \sqrt{(Z-ZINLT)^2 + (|Y|-YINLT)^2}$$

In the second region outboard of the inlet and under the wing, the polar location is found between the two closest traverses generated radially from the inlet corner, XINLT,YINLT,ZINLT. The axial location is computed for each polar traverse in XVSR. The local shock location is then computed by linear interpolation in polar angle, ϕ. The parameter β_{IS} is computed as $\beta_{IS} = XS/RS$.

For the remainder of the flow field, Region III (see sketch in Section A-20), the shock location and β_{IS} are generated from free stream values. The shock location is thus set to $XS = \beta_\infty RS$. In each of the regions, the computed shock shape is assumed to contain any angle of attack corrections and no further rotations are performed at this point.

The descriptions of the parameters in the argument list follow:

BETAI	β_{IS}, Mach number parameter for first influence of inlet panels at Y,Z
XS	axial location of inlet shock at location Y,Z

RS radial distance from closest inlet to point Y,Z

For $Y \geq YCPI$, distance in plane of traverse

For $Y < YCPI$, distance from inlet inboard leading edge

Y,Z lateral location of point at which inlet shock is computed

C-43 Function INLTST

Logical function INLTST determines whether the panel index in the argument is for an inlet panel. The value of the function is set to TRUE if the index is for an inlet panel and FALSE if it is not. In addition the logical variable OPEN is set to indicate whether the inlet panel is blocked or unblocked to flow. Three conditions are tested for. If no inlet panels exist INLTST and OPEN are set false. If an inlet panel exists the index I is compared to the table of possible inlet panel numbers, JINLT. If I is equal to one of the inlet panel numbers INLTST is set true. If I is not an inlet panel number INLTST is set false. If I falls in the subset of JINLT of unblocked panels, OPEN is true; otherwise OPEN is set false. A listing of this routine is presented in Figure C-1(dd) of this report.

The descriptions of the parameters in the argument list follow:

I panel number index to be compared with table of possible inlet panel numbers, JINLT

OPEN logical variable indicating whether an inlet panel is open or blocked. OPEN is TRUE if panel allows unblocked flow through panel

Called by:

FLDVL2

C-44 Subroutine INTOST

Subroutine INTOST (see Figure C-1(dd) for a listing) takes a vector with components specified in the inertial ξ,η,ζ coordinate system directions and transforms it into a vector with components in the store x,y,z coordinate system directions, see Figure 21 of Reference 1. That is

$$\begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} = [A]^{-1} \begin{pmatrix} s_\xi \\ s_\eta \\ s_\zeta \end{pmatrix}$$

The matrix $[A]^{-1}$ is the transpose of the direction cosine matrix given by Equation (B-2) of Appendix B of Reference 2. The transpose is equal to the inverse since [A] is orthogonal. The matrix [A] is either calculated in subroutine DIRCOS for the separating store or in RESVEL for fixed stores.

In terms of the above notation, the quantities in the parameter list of the subroutine are:

XI	s_ξ
ETA	s_η
ZETA	s_ζ
X	s_x
Y	s_y
Z	s_z
DC	[A]

Called by:

VXYZ, SFORCE, SEMFOR, SFORC2, DEMON2, EJECTR, RESVEL

C-45 Subroutine INVER2

Subroutine INVER2 solves the system of simultaneous linear algebraic equations.

$$[A]X = B$$

This routine performs pivot searching during the solution of the general matrix, A. The right hand side is passed into INVER2 as the N+1'st column in the matrix. The solution, X, also returns in that location. A listing of this routine is presented in Figure C-1(dd) of this report. The routine is currently limited to 6 equations by internal dimensions. The descriptions of the parameters in the argument list follow:

A	coefficients of linear system of equations in first N columns; columns N+1 through N+NSYS contain multiple right-hand sides, B, on input and solutions, X, on return
NSYS	number of right hand sides
N	actual number of equations
NMAX	first dimension of A
MMAX	second dimension of A

If the coefficient matrix is found to be singular an error message is printed out (see Section 4.4 of Volume II) and the program stops.

Called by:
    TRJTRY

C-46 Subroutine IOREAD

Subroutine IOREAD performs an unformatted read from external file, IO. NA consecutive elements of array, A, are read sequentially. This routine is used to specify a common interface to external files. A listing of this routine is presented in Figure C-1(ee) of this report. The descriptions of the parameters in the argument list follow:

IO	external file reference number
A	array of numbers to be read
NA	number of element of A to be read

A machine dependent version of IOREAD is available for CDC machines using asynchronous input routine BUFFER IN.

Called by:

EGMRST, FRSTRT, SFORC2, SMARCH, SOLVUV, DEMON2, TRJTRY, FLDUVW

C-47 Subroutine IOWRIT

Subroutine IOWRIT performs an unformatted write to external file, IO. NA consecutive elements of array, A, are written sequentially. This routine is used as a common interface to external files. A listing of this routine is presented in Figure C-1(ee) of this report. The descriptions of the parameters in the argument list follow:

IO	external file reference number
A	array of numbers to be written
NA	number of element of A to be written

A machine dependent version of IOWRIT is available for CDC machines using asynchronous output routine BUFFER OUT.

Called by:

EGMSAV, FRSTRT, STRDAT, FLDAC2, CRFWBD, DEMON2

C-48 Subroutine IXBOD

Subroutine IXBOD scans the elliptic store geometry to locate and define leading and trailing edges of body-fin sections and the elliptic body axes for the interference shell. This routine uses the parameters provided in input item 16 of Program II to search the geometry arrays passed from Program I. The indices of the panels bounding each of the body alone and finned-body sections are first determined. For each finned section the leading edge and the maximum vertical and horizontal elliptic semi-axes are determined. The program logic is limited to handling one body segment at a time. A listing of the routine is presented in Figure C-1(ee). The descriptions of the parameters in the argument list follow:

XBOD(I)	trailing edges of Ith body section; see input item 16
IBOD(I)	ring index computed for Ith body section
NBOD	number of body sections; see input item 16
LFIN(I)	logical section type indicator of Ith section; see input item 16
XWLE1	axial station of first finned-body section leading edge
XWLE2	axial station of second finned-body section leading edge
RA1	maximum vertical semi-axis of first finned-body section

RA2 maximum vertical semi-axis of second finned-body section

RB1 maximum horizontal semi-axis of first finned-body section

RB2 maximum horizontal semi-axis of second finned-body section

Called by:

TRJTRY

C-49 Subroutine LAYBIP

Subroutine LAYBIP lays out and determines the geometrical properties of the constant u-velocity panels on the elliptic store body where mutual interference occurs. This routine uses the meridional angles specified in Program I for the elliptic body to determine the panel edges for the u-veocity panels. It guarantees that the interference shell panel lay out matches the same meridional spacing as the source panels. The routine performs an outer loop over the number of panels around the circumference and an inner loop over the number of panels axially. The geometric panel properties are stored in the arrays in labeled common ONE (see descriptions in Section D-2 of Appendix D). A listing of the routine is presented in Figure C-1(ff) of this report. A description of the parameter in the argument list follows:

NBDCR number of panels around the circumference of the interference shell. See input item 18 of Program II.

Called by:

CRFWBD

C-50 Subroutine LAYOUT

Subroutine LAYOUT lays out and determines the geometrical properties of the constant u-velocity panels on a single fin surface. The fin external shape may be trapezoidal or may have multiple breaks in leading and trailing edge sweep angles. A listing of the routine is presented in Figure C-1(ff) of this report. This routine is adapted from the work of Reference 5 and specialized to fins only. The output variables from this routine are stored in labeled common ONE, and are defined in Section D-2 of Appendix D.

The descriptions of the parameters in the argument list follow:

SLPWLE	fin leading edge sweep for trapezoidal planform description
SLPWTE	fin trailing edge sweep for trapezoidal planform description
Y(I)	Y stations of parameters of variable sweep description
MSWP	number of spanwise panels to be laid out on fin
CRP	fin root chord
NS	fin quadrant indicator NS=1, right or upper right interdigitated fin NS=2, left or lower left interdigitated fin NS=3, upper or lower right interdigitated fin NS=4, lower or upper left interdigitated fin

CTP panel tip chord

PHI dihedral angle of panel; measured counterclockwise from positive Y-axis

THET meridional angle of intersection of fin with body; counterclockwise from positive Y-axis

Called by:
CRFWBD

C-51 Subroutine LOADS

Subroutine LOADS calculates the forces and moments on the elliptic store fins and the interference shell. The panel forces are first computed in the panel coordinate system and then transformed to the empennage system. A listing of the routine is presented in Figure C-1(hh) of this report.

The descriptions of the equations used to compute the forces and moments for interdigitated and cruciform fins are presented in Appendix F of Reference 5. The method has been generalized, here, to handle other fin arrangements. The descriptions of the normal force calculation for the elliptic body is similarly presented in Section 4.2 of the same reference. All forces are normalized by the reference area. All moments are normalized by both the reference area and reference length. The sense of the resulting coefficients are as follows:

C_Y acts along empennage system Y-axis, positive to the right

C_Z acts along empennage system Z-axis, positive up

C_m moment about empennage system Y-axis, positive nose up viewed from the rear

C_n moment about empennage system Z-axis, positive nose to right viewed from the rear

C_ℓ moment about empennage system X-axis, positive clockwise viewed from the rear

The description of the parameter in the argument list is:

HEAD alphanumeric heading; see input item 13

Subroutine references:
ROTBW, ROTFW, SPNLD

Called by:
SPECPR

C-52 Subroutine NUMACH

Subroutine NUMACH determines the local Mach numbers at the wing leading and trailing edges at a given field point lateral and vertical position relative to the wing. The calculations are performed for wing thickness only. After the Mach numbers are determined the calculations are repeated using these Mach numbers to determine the leading and trailing edge influence locations. The method used is described in Section 4.3 of Reference 2. A listing of the program is presented in Figure C-1(ii) and a flow chart in Figure C-8 of this report.

At the beginning of the subroutine a test is performed to determine if the specified z_w location (Z) is below the wing. If not, an error message is printed (see Section 4.4 of Volume II) and the program stops.

Next, the subroutine calculates the local wing chord, CHRD, at the specified y_w location (Y). Quantities previously determined by subroutine WLYOUT in Volume III are used in this

calculation. As shown in the flow chart, if the field point is below the fuselage, CHRD equals the wing root-chord; outboard of the wing tip CHRD equals the tip-chord.

After initializing certain control variables, the subroutine next begins the calculation of an x traverse aft of $x_w = 0$. At each point the velocity due to wing thickness is calculated (subroutine VELWT2). The point, XSAVE, at which wing thickness induces the maximum turning angle, $\Delta\nu$, is determined using an x-interval size, DELX1, equal to one-hundredth CHRD. The search is refined by backing up to XSAVE-DELXI and repeating the calculations using DELX1 equal to one-tenth the previous interval size. Once the point in the traverse is isolated (XT3,Y,Z) at which the turning angle is a maximum, the local Mach number, M_ℓ (FMXT3), specified by Equations (44), (45), (46), and (61) of Reference 2 is computed from the velocity components.

The search for XT3 described above is next repeated with velocity calculations based on M_ℓ (specified when INUMCH=1). The new value of XT3 defines the beginning of the range of influence associated with the wing thickness panels. The remainder of the subroutine performs a similar calculation to determine the local Mach number and location associated with the wing-thickness trailing-edge influence. The search starts at seven-tenths of the local wing chord. The notation for the Mach number and the trailing edge influence location are FMXT4 and XT4, respectively.

The descriptions of the parameters in the argument list follow:

Y,Z	coordinates of field point in wing coordinate system

Subroutine references:

VELWT2

Called by:

SFORCE, SFORC2

C-53 Subroutine PANVEL

Subroutine PANVEL organizes the calls to SORPAN for the calculation of the influence of a source panel at a field point. This routine computes the transformations and rotations necessary to calculate the influence of an arbitrarily oriented panel at either a field point or at the control point of another arbitrarily oriented panel. A listing of the routine is presented in Figure C-1(kk) of this report.

The routine first computes the combined transformations to rotate a panel or its image into the local panel system and back. If the same transformation is to be used for a number of calculations the logical variable, LZERO, is used to skip around this code in subsequent calls. The field point in question is rotated into the panel coordinate system and a call made to SORPAN to compute the influence coefficient. If the body is symmetric (IXZSYM=0) the above calculations are repeated for the symmetric panel on the opposing side of the X-Z plane. The influence is summed and rotated back into the original body coordinate system.

The descriptions of the parameters in the argument list follow:

UB,VB,WB	component velocity influence coefficients in body coordinate system at field point
AN	normal velocity influence coefficient of a panel at another arbitrarily oriented panel control point; set equal to WB for a field point
IXZSYM	body X-Z plane symmetry option

Subroutine references:

SORPAN

Called by:

FLDAC2, FLDVL2

C-54 Subroutine PAS001

Subroutine PAS001 performs the [L*U] decomposition of the positive definite matrix, A. This routine performs no pivot search during decomposition. It does skip unnecessary calculations associated with off-diagonal zeroes. An error return is provided for encountering zero values on the diagonal. The decomposition procedure is equivalent to

$$A = \begin{bmatrix} 1.0 & & & & \text{zero} \\ L_{21} & 1.0 & & & \\ . & & . & & \\ . & & & 1.0 & \\ L_{n1} & . & . & L_{nn}-1 & 1.0 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & . & . & u_{1n} \\ & u_{22} & & & . \\ & & . & & . \\ & & & . & . \\ & \text{zero} & & & u_{nn} \end{bmatrix}$$

The decomposed matrix is stored in the location of the original matrix. A listing of the routine is presented in Figure C-1(kk) of this report.

The descriptions of the parameters in the argument list follow:

A	positive definite matrix with nonzero diagonal elements
N	actual size of matrix, A
NER	diagnostic. If NER greater than zero, decomposition failed because $\lvert A(NER,NER) \rvert < 10^{-20}$
ND	dimensioned size of A

Called by:

CRFWBD

C-55 Subroutine PAS002

Subroutine PAS002 solves the system of equations [L*U] * X = B by forward and backward substitution. It is assumed that matrix, A, has been decomposed by routine PAS001 or equivalent so that A contains [L*U]. A listing of the routine is presented in Figure C-1(11) of this report.

The descriptions of the parameters in the argument list follow:

A	coefficient matrix containing [L*U]
B	matrix containing right-hand sides of the linear system [L*U] * X = B. Contains X on output
N	actual size of [L*U] matrix contained in A
NB	number of right-hand side vectors contained in the first NB columns of B
NDA	dimensioned size of A
NDB	dimensioned size of B

Called by:

DEMON2, SMARCH

C-56 Subroutine PITROL

Subroutine PITROL computes the crossflow components of velocity due to an elliptic body subjected to a nonuniform flow field $V_{AV}(x)$, $W_{AV}(x)$ at the centerline. The routine first interpolates in the table of average nonuniform flow field velocities versus axial station to find the local flow the body is subjected to. Following this the velocities at field point X,Y are computed. The equations and transformation in the complex plane used to compute the crossflow components are described in Section 6 of Appendix I of Reference 5. A listing of the routine is presented in Figure C-1(11) of this report. The descriptions of the parameters in the argument list follow:

XLOC — axial station in source panel coordinate system of cross section

VXAV,WXAV — average nonuniform flow field velocity components at cross section

X,Y — lateral coordinates (Y,Z in body coordinates) of point at which cross flow velocities are computed

VS,WS — crossflow velocity components at X,Y

Subroutine references:

DSDZ, Z

Called by:

F

C-57 Subroutine PRESS

Subroutine PRESS computes the source panel pressure coefficient using the exact Bernoulli pressure formula. The pressure coefficient is limited by the vacuum pressure coefficient, $C_p = -p_\infty/q_\infty$. Given the u,v,w velocities at each source panel control point, the routine performs a single loop over the number of panels to compute pressure coefficients. A listing of the routine is presented in Figure C-1(11) of this report. The descriptions of the parameters in the argument list follow:

NBODY — number of source panels

U,V,W — arrays containing the velocity components in the body fixed coordinate system used with the source panels

CPP — array containing computed pressure coefficients

CPSTAG limiting stagnation pressure coefficient

CPCRIT critical pressure coefficient

CPVAC vacuum pressure coefficient

Called by:
SDSTN2

C-58 Subroutine RDFILE

Subroutine RDFILE reads the file produced by Program I which contains all of the data read into Program I or computed in that program which is used by Program II. This routine reads the information written on TAPE12 in Program I into the appropriate arrays in this program. When reading the arrays generated for the noncircular configurations, all data for the fuselage are copied onto TAPE11. Similarly all data for the elliptic stores are copied onto TAPE10. The data are stored there until such time as the separating store has been specified. A listing of the routine is presented in Figure C-1(mm) of this report. The arrays into which the data is read by this routine are described in Section D-2 of Appendix D for the appropriate labeled common blocks.

Subroutine references:
CRSTRT

Called by:
TRJTRY

C-59 Subroutine REFSHK

Subroutine REFSHK calculates the location at which the circular store nose shock which is reflected by the wing strikes

AD-A099 332 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 20/4
PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLU--ETC(U)
NOV 80 J MULLEN, F K GOODWIN, M F DILLENIUS F33615-76-C-3077
UNCLASSIFIED NEAR-TR-210-VOL-4 AFWAL-TR-80-3032-VOL-4 NL

2 OF 4
AD A099332

the store and the value of β to be used for the image store. The equations used to compute the reflected shock influence on the circular store are described in Section 4.4.1 of Reference 2. A listing of the routine is presented in Figure C-1(mm) of this report.

The steps used in this routine to compute the reflected shock location are as follows:

1. Calculate store nose location in inertial (fuselage) system $(x_{B_n}, y_{B_n}, z_{B_n})$
2. Calculate store nose location in wing system $(x_{w_n}, y_{w_n}, z_{w_n})$
3. Assume longitudinal axis of store shock wave is parallel to wing root chord and calculate x_{shock} at $r = z_{w_n}$
4. Locate shock intersection point in wing system $(x_{w_s}, y_{w_s}, z_{w_s})$
5. Calculate local $x_{w_{\ell e}}$ and $x_{w_{te}}$ at y_{w_s}
6. If x_{w_s} is between $x_{w_{\ell e}}$ and $x_{w_{te}}$ integrate wing thickness slope distribution up to x_{w_s} to get wing half thickness at this point, t/2

$$\frac{t}{2} = \frac{x_{w_{\ell e}} - x_{w_{te}}}{NCWS} \sum_{n=1}^{\left[\frac{(x_{w_{\ell e}} - x_{w_s})}{(x_{w_{\ell e}} - x_{w_{te}})}\right] NCWS + 1} \left(\frac{dz}{dx}\right)_n$$

7. Calculate x_{shock} at $r = z_{w_n} - t/2$ and locate in wing system $(x^*_{w_s}, y^*_{w_s}, z^*_{w_s})$

8. Calculate shock angle at $x^*_{w_s}$

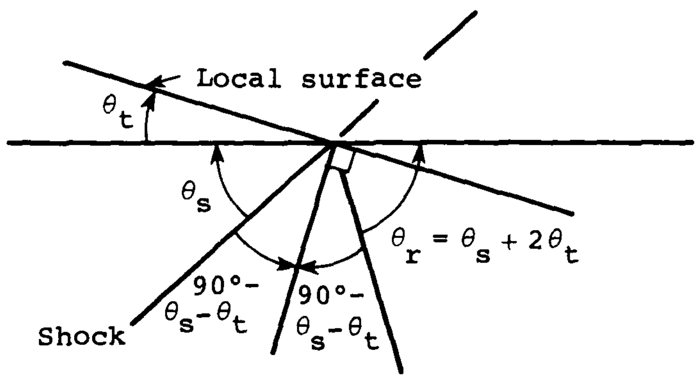
$$\theta_s = \tan^{-1} \frac{\Delta r}{\Delta x} \quad \text{from shock shape table}$$

9. Determine thickness slope at $x^*_{w_s}$

$$\theta_t = \tan^{-1} \left(\frac{dz}{dx}\right)_n \qquad n = \left(\frac{x_{w_{\ell e}} - x^*_{w_s}}{x_{w_{\ell e}} - x_{w_{te}}}\right) \text{NCWS} + 1$$

10. Determine shock reflection angle

$$\theta_r = \theta_s + 2\theta_t$$



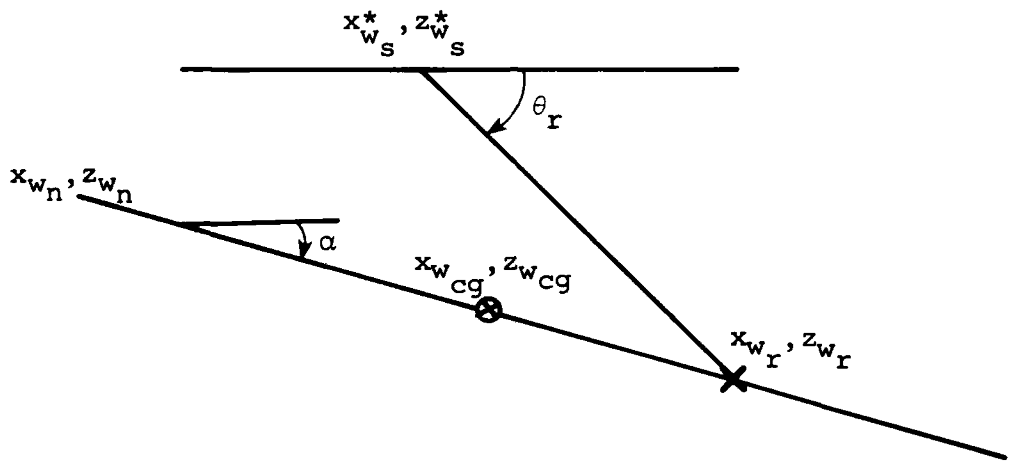
11. Locate store c.g. in wing system assuming $y_{w_{cg}} = y_{w_n}$

$$x_{w_{cg}} = \mathrm{VAR(7)} - \mathrm{XBWOC}$$

$$y_{w_{cg}} = y_{w_n}$$

$$z_{w_{cg}} = \mathrm{VAR(9)} - \mathrm{ZBWO}$$

12. Determine intersection of reflected shock wave and line from store nose through c.g.



$$z^*_{w_s} = t/2$$

$$\alpha = \tan^{-1}\left(\frac{z_{w_{cg}} - z_{w_n}}{x_{w_n} - x_{w_{cg}}}\right)$$

If $\theta_r \le \alpha$ there is no intersection. For $\theta_r > \alpha$, the intersection of the reflected shock with store axis is determined from the intersection of the following two equations.

Equation of reflected shock

$$z_{w_r} = z^*_{w_s} + \tan\,\theta_r\,(x^*_{w_s} - x_{w_r})$$

Equation of store axis

$$z_{w_r} = z_{w_n} + \tan\,\alpha\,(x_{w_n} - x_{w_r})$$

After subtracting the two equations, the point at which the reflected shock intersects the store axis is

$$x_{w_r} = \left(\frac{1}{\tan\alpha - \tan\theta_r}\right)\left(z_{w_n} - z^*_{w_s} + x_{w_n}\tan\alpha - x^*_{w_s}\tan\theta_r\right)$$

$$z_{w_r} = z_{w_n} + \tan\,\alpha\,(x_{w_n} - x_{w_r})$$

If $\theta_r = 90°$

$$x_{w_r} = x^*_{w_s}$$

13. Calculate distance from nose measured along store axis to reflected shock

$$|x_{s_r}| = -(x_{w_r} = x_{w_n})\cos\,\alpha + (z_{w_s} - z_{w_n})\sin\,\alpha$$

If $|x_{s_r}|$ is greater than store length, there is no intersection.

The descriptions of the parameters in the argument list follow:

NS	number of values in shock shape table
SX,SR	arrays containing the tabulated values of X and R describing the shape of the shock striking the wing, including any rotation with store angle of attack

Subroutine references:
    STTOIN

Called by:
    SFORCE

C-60 Subroutine RESVEL

Subroutine RESVEL, along with the other subroutines which it calls, calculates the perturbation velocities, as described in Section 5 of Reference 2, in the inertial coordinate system at a specified field point due to all aircraft components other than the separated store and its image. A listing of the routine is presented in Figure C-1(nn) and a flow chart in Figure C-9 of this report. The descriptions of the variables used by this routine which are in labeled common are found under the descriptions of common variables in Section D-2 of Appendix D.

The velocity contributions of each of the parent aircraft components are computed in order. The β's used by each of the components are assumed to have previously been computed by SFORCE

or SFORC2 and BVARIA. The influence of the fuselage is computed first. If a circular fuselage is present, NFU=1, velocities induced by the source and doublet distributions are calculated using subroutine VELCAL. If the noncircular fuselage is present, NFU=2, the influence of the source panels is computed in routine FLDVEL. If a ramp inlet is present on the noncircular fuselage, the fixed point is checked to determine whether it is in the influence of the inlet shock. Figure C-10 shows a typical shock region of influence beneath the inlet for partially blocked flow with a mass flow ratio of about 0.4. The inlet shock influence is assumed to be felt between the two lines represented as β_{IS} for the inlet shock location and β_∞ for the inlet shoulder or aft end of the inlet influence. For the radial location of each field point, the axial location, XIS, and the corresponding β_{IS} of the inlet shock are computed in routine INLBET. If XIS is less than XBTINL, the point is considered in Region I. If the point lies forward of XIS or aft of XISHLD, the β computed from the fuselage nose shock in BVARIA is used and the velocities computed in FLDVEL. If the point lies between XIS and XISHLD, XMAP is set to -XB and

$$\beta = \max\left[0.01,\ \beta_\infty + \left(\frac{\mathrm{XIS - XCLSD}}{\mathrm{RIS - RCLOSD}} - \beta_\infty\right)\frac{\mathrm{DX}}{\mathrm{DXS}}\right]$$

If XIS is greater than or equal to XBTINL, the point is considered in Region II. If the point is outside the inlet influence, velocities are computed as before. If the point lies between XBTINL and XISHLD, the points between XIS and XISHLD must be mapped to an equal position between XBTINL and XISHLD. This is necessary to match the first influence from the panels on the inlet face propagating along the β_I line. Thus XMAP and β are set to the following and velocities computed in FLDVEL.

XMAP = XINLT+XISHLD-DX*DXBT/DXS

and

$$\beta = \begin{cases} \max\left[0.01,\ \beta_\infty + \left(\dfrac{XIS - XCLSD}{RIS - RCLOSD} - \beta_\infty\right)\dfrac{DX}{DXS}\right], & XBP > XIS \\ BTINLT & , XBP \leq XIS \end{cases}$$

To compute the interference shell influence, the point is translated into the wing coordinate system, and any induced velocities computed by VELBD2. Next, the wing constant u-velocity panel and thickness panel velocities are calculated using subroutines VELWP2 and VELWT2, respectively. If a pylon is present, NPY=1, the pylon velocities are calculated by calling VELPP2 for constant u-velocity panels and VELPT2 for thickness panels. If a rack is present, NRACK=1, the nose of the rack is located relative to the fuselage and any induced velocities computed in VELCAL.

If additional stores are present, NSTRS greater than one, the induced influence is computed by first locating the point relative to the fixed store body coordinate system and calculating velocity contributions. If the fixed store is circular, NSHAPE(J) less than 51, the velocities are computed by VELCAL. If the fixed store is elliptic, NSHAPE(J) greater than 50, the velocities are computed in FLDVEL. The velocities computed in the local store coordinates must then be rotated back into the inertial system and added to the other components.

The descriptions of the parameters in the argument list follow:

XB,YB,ZB — coordinates in inertial system of point at which velocity field is to be calculated

UTU,VTV,WTW — sum of u/V_∞, v/V_∞, w/V_∞ perturbation velocities due to parent aircraft in inertial coordinate system

Subroutine references:

FLDVEL, INLBET, VELBD2, VELCAL, VELPP2, VELPT2, VELWP2, VELWT2, INTOST, STTOIN

Called by:

SFORCE, SEMFOR, SFORC2, DEMON2

C-61 Subroutine ROTBW

Subroutine ROTBW transforms velocities from the body interference panel coordinate system to the empennage coordinate system. The transformation performed is:

$$Y_o = Y_i \cos \theta_2 + Z_i \sin \theta_2$$
$$Z_o = -Y_i \sin \theta_2 + Z_i \cos \theta_2$$

where θ_2 is angle of the panel normal measured from the vertical. It is shown in Figure 18 of Reference 1 depicted as $\theta_{2,BIP}$. A listing of the routine is presented in Figure C-1(pp) of this report. The descriptions of the parameters in the argument list follow:

YI,ZI — input lateral and vertical velocity components in body interference panel coordinate system to be transformed

YO,ZO — output transformed velocities in empennage coordinates

I — index of body interference panel number

Called by:
VELNOR, LOADS

C-62 Subroutine ROTFW

Subroutine ROTFW transforms velocities from the local fin coordinate system to the empennage reference coordinate system. The transformation performed is:

$$Y_{OUT} = Y_{IN} \cos \phi_F - Z_{IN} \sin \phi_F$$

$$Z_{OUT} = Y_{IN} \sin \phi_F + Z_{IN} \cos \phi_F$$

where ϕ_F is the angle of the panel normal to the vertical. It is shown in Figure 18 of Reference 1 depicted as ϕ_F. A listing of the routine is presented in Figure C-1(pp) of this report. The descriptions of the parameters in the argument list follow:

YIN,ZIN	input lateral and vertical velocities in fin coordinate system to be transformed
YOUT,ZOUT	output transformed velocities in empennage coordinates
PHIF	angle of rotation of plane of fin from horizontal, radians

Called by:
VELNOR, SPECPR, LOADS

C-63 Subroutine ROTWB

Subroutine ROTWB transforms velocities from the empennage reference coordinate system to the body interference panel coordinate system. The transformation performed is:

$$Y_o = Y_i \cos \theta_2 - Z_i \sin \theta_2$$

$$Z_o = Y_i \sin \theta_2 + Z_i \cos \theta_2$$

where θ_2 is the angle of the panel normal from the vertical. It is shown in Figure 18 of Reference 1 depicted as $\theta_{2,BIP}$. A listing of the routine is presented in Figure C-1(pp) of this report. The descriptions of the parameters in the argument list follow:

YI,ZI	input lateral and vertical velocity components in empennage reference coordinates to be transformed
YO,ZO	output transformed velocities in body interference panel local coordinate system
I	index of body interference panel number

Called by:
CRFWBD, VELNOR

C-64 Subroutine ROTWF

Subroutine ROTWF transforms velocities from the empennage coordinate system to the local fin coordinate system. The transformation performed is:

$$Y_{OUT} = Y_{IN} \cos \phi_F + Z_{IN} \sin \phi_F$$

$$Z_{OUT} = -Y_{IN} \sin \phi_F + Z_{IN} \cos \phi_F$$

where ϕ_F is the angle of the panel normal measured from the vertical. It is shown in Figure 18 of Reference 1 depicted as ϕ_F.

A listing of the routine is presented in Figure C-1(qq) of this report. The descriptions of the parameters in the argument list follow:

YIN,ZIN	input lateral and vertical velocity components in the empennage coordinate system to be transformed
YOUT,ZOUT	output transformed velocities in the fin coordinate system
PHIF	angle of rotation of plane of fin from horizontal, radians

Called by:

CRFWBD, VELNOR, DEMON2, SPECPR

C-65 Subroutine SDSTN2

Subroutine SDSTN2 organizes the solution for source panel strengths and calculation of the forces and moments for the separating elliptic store including the effects of the image store. All the calculations in this routine are performed in the store source panel coordinate system (x_s, y_s, z_s) shown in Figure 13 of Reference 1. A listing of this routine is presented in Figure C-1(qq) of this report.

SDSTN2 is organized to perform a series of calls to different routines to setup and calculate the store boundary condition, solve for the panel strengths, compute the resulting velocities and pressures on the panel surfaces, and sum the forces and moments. The separating store boundary condition is first set equal to the normal component of the nonuniform flow field computed at the panel control points in routine SFORC2. The normal is computed from the component velocities and the panel inclination and

incidence angles in routine VNORM. The solution for the panel strengths including the effects of the image store is then computed in routine SMARCH using a ring by ring marching solution.

The velocities used in the pressure calculation are the sum of the perturbation velocities and the contributions of damping terms due to p, q, and r motion of the store. The perturbation velocities are computed in SOLVUV from the panel strengths and the U, V, and W matrices stored on TAPE8 as passed from Program I. The damping terms are added in routine VELDMP. The Bernoulli pressure acting at the control point of each panel is then computed by PRESS from the above velocity components. The forces and moments and the distributions of loads on the body are lastly computed in FORMOM based on the above pressure acting at the control points of the panels.

The descriptions of the parameters in the argument list follow:

UT,VT,WT — arrays containing the three components of velocity in the store source panel coordinate system due to all parent aircraft and free stream effects for the translating and rotating store

LSKP(I) — logical array used to hold temporary calculation test variable for Ith panel

IMAGE — logical indicator whether image store effects are to be computed
 IMAGE=.true., image effects computed
 IMAGE=.false., no image store influence computed

Subroutine references:
 VNORM, SMARCH, SOLVUV, VELDMP, PRESS, FORMOM

Called by:
 SFORC2

C-66 Subroutine SDSTRN

Subroutine SDSTRN calculates the strengths of the line sources and doublets modeling the separating store in the nonuniform flow field. This routine is called by SFORCE when the circular store option is used. The singularity strengths are calculated using the equations in Section A-2 of Appendix A of Reference 2. A listing of the routine is presented in Figure C-1(qq).

At the beginning of the routine the free-stream axial and total Mach numbers as seen by the store are calculated. If a wing image store is present, IMSTOR≠0, the image store is located relative to the real store and other quantities used in the image store velocity field calculation are determined. If a fuselage image store is present, IMFSTR≠0, similar quantities are calculated.

The remainder of the routine calculates the strengths of the line sources and doublets modeling the separating store in the aircraft flow field, including image store influences, if any. The strengths of the source and doublets originating at the store nose are first calculated. The remaining singularity strengths are calculated in a DO loop over the control points.

At the beginning of the loop the influence of upstream sources and doublets at the control point is calculated. The next section of the routine is executed if a wing image store is present, IMSTOR≠0. The values of β_a and β_s are determined and subroutine IMSVEL is called to calculate the wing image store induced velocities. They are added to the previously calculated parent aircraft and free stream velocities at the control points.

The next part of the routine repeats the above calculation for the fuselage image stores if they are present, IMFSTR≠0. The values of β_a and β_s are calculated for the image store and the

centerline image store. Two calls are then made to IMSVEL to calculate the velocities induced by these stores. The velocities due to the image store are added to the previously calculated totals and those due to the centerline store are subtracted.

The last section of this subroutine calculates the strengths of the nth source and doublets using the previously calculated velocities in the boundary condition.

The parameter in the subroutine argument list is defined as follows:

XSHOL	location of store shoulder, point of maximum radius, measured from store nose

Subroutine references:
IMSVEL, SDTRMS

Called by:
SFORCE

C-67 Subroutine SDTRMS

Subroutine SDTRMS evaluates the square root and inverse hyperbolic cosine terms appearing in the line source and doublet expressions. The expressions computed are:

$$\mathrm{ROOT(A)} = \begin{cases} \sqrt{A^2 - 1} & , \ A > 1 \\ 0 & , \ A \leq 1 \end{cases}$$

$$\mathrm{ACOSH(A)} = \begin{cases} \ell n \ (A+\mathrm{ROOT(A)}), & A > 1 \\ 0 \qquad , & A \leq 1 \end{cases}$$

A listing of the routine is presented in Figure C-1(rr) of this report.

Called by:
SFORCE, SDSTRN, IMSVEL

C-68 Subroutine SEMFOR

Subroutine SEMFOR calculates the empennage forces and moments by the method described in Section 5.3 and Appendix I of Reference 6. A listing of the subroutine is presented in Figure C-1(ss), and a flow chart in Figure C-11.

An examination of the flow chart shows that the first steps in the routine are to locate the point at which the empennage forces act relative to the store moment center and to set JMAX equal to 2 or 4 depending on whether the empennage is planar or cruciform.

The next part of the subroutine calculates the perturbation velocity field at the MSF control points on each of the JMAX fins. After the parent aircraft velocities are calculated by subroutine RESVEL they are resolved into the store-body coordinate system by subroutine INTOST. The image store induced velocities are next calculated and added. The free-stream components are then calculated, resolved into the store-body coordinate system, and added to the perturbation velocities. The resultant velocities are made dimensionless by the store free-stream velocity. If aerodynamic damping is being included, the pitch and yaw damping terms are then added. From these velocities the components normal to the fin surfaces are determined. Positive directions are shown in Figure 10 of Reference 6.

The remainder of the routine calculates the empennage forces and moments. First W_O and V_O shown in Figure 10 of Reference 6 are determined in a manner similar to that used for the fin

control points. Then, the normal force and the side force, if the empennage is cruciform, are calculated using Equations (I-13) and (I-18) of Reference 6. The spanwise integrations are performed using Simpson's rule. It is to be noted that in the present program all four fins are assumed to have the same span, $s_h = s_v$. The pitching moment and yawing moment are calculated using Equations (I-21) and (I-22).

These forces and moments are in the fin coordinate system. They are resolved into the body coordinate system using equations (58) through (61) of Reference 6.

Finally, if rolling moment is to be calculated this is done using Equations (I-30) or (I-52) of Reference 6. Equation (I-30) is for a planar empennage and (I-52) is for a cruciform empennage.

Subroutine references:

INTOST, RESVEL, SIMSON, STTOIN, ZIMAGE

Called by:

TRJTRY

C-69 Subroutine SEMPIN

Subroutine SEMPIN initializes certain quantities which will be used repeatedly in the circular store option empennage force and moment calculation, subroutine SEMFOR. The equations programmed are given in Appendix I of Reference 6. A listing of the subroutine is presented in Figure C-1(tt) and a flow chart in Figure C-12 of this report.

The flow chart indicates that the first calculation performed is to determine the radial distance outward from the body axis to the MSF fin control points. The first point is at the body-fin

juncture, $r_f = a$ (see Figure 10, Reference 6), and the last is at $r_f = s_h = s_v$. The others are equally spaced in between these two points. Next, a check is made to determine that XTAIL was input as a negative quantity and then the angular orientation of the fins in the store-body coordinate system is determined. Referring to Figure 10 of Reference 6, these angles are measured in the clockwise direction from the z_s axis.

JMAX is next set equal to 2 or 4 depending on whether the empennage is planar or cruciform and then the y and z coordinates of the control points on all of the fins are determined. Next, certain constants are calculated and then the values of $(cc_\ell)_3$ are calculated at the control points. They are the same for all panels since $s_h = s_v$ (see Equations (I-14) and (I-19), Reference 6).

If rolling moment is not to be calculated, NROLL=0, control is returned to the calling program. If rolling moment is to be calculated, and the empennage is planar, IPLNR=1, $(cc_\ell)_5$ given by Equation (I-29) of Reference 6 is calculated at the control points. Note that in the program the following substitution is made

$$\cosh^{-1}(x) = \ell n(x + \sqrt{x^2 - 1})$$

For a cruciform empennage, IPLNR=0, Equation (I-51) of Reference 6 is used for the first control point where $y_f = a$. For the other control points Equation (I-38) is used. The following substitution is made in the program

$$\tanh^{-1}(x) = \frac{1}{2} \ell n \frac{(1 + x)}{(1 - x)}$$

Subroutine references:

CEL1, CEL2, ELI1, ELI2

Called by:
TRJTRY

C-70 Subroutine SFORCE

Subroutine SFORCE calculates the aerodynamic forces and moments acting on the separating store when the circular store option of the computer program is used, see Section 6.2.1 of Reference 2. The forces and moments are calculated using the equations in Section A-3 of Appendix A of Reference 2. A listing of the routine is presented in Figure C-1(uu). A flow chart is shown in Figure C-13.

At the beginning of the routine the Mach numbers and Mach cone angles to be used in the source and doublet determination are calculated. Next, a test is performed to determine if, at the base of the body, the radial distance to the β_s Mach cone emanating from the body nose is less than the maximum radius of the body. If so, an error message is printed out (see Section 4.4 of Volume II) and the program stops. If this test is passed, the origins of all of the line singularities are calculated.

The next sections of the subroutine are devoted to revising the layout of the body definition points and control points and the origins of the line singularities. If the first control point lies outside the Mach cone from the nose an iteration is performed to determine the intersection of the cone with the body surface. Once this point is found the body definition points are redistributed over the remainder of the body. Next, the axial locations of the control point and the body radius and surface slope at these points are calculated. The origins of the line singularities are then defined.

Subroutine NUMACH is next called to calculate the locations of the wing leading-edge and trailing-edge shock waves and the Mach numbers associated with these two points. The points at which the separated store nose shock wave, which reflects off the wing and fuselage, strike the separated store are determined with successive calls to REFSHK and FREFSH.

The next three loops determine the velocity field in which the store is immersed. The first loop locates each control point in the fuselage, or inertial, coordinate system; calls BVARIA to calculate the Mach cone angles, betas, to use in the fuselage, rack, and stores velocity calculations; calculates the Mach number and betas to use in calculating the wing induced velocities; and calls RESVEL to calculate the parent aircraft induced velocities. Upon returning from RESVEL, these velocities are resolved into the store coordinate system by calling INTOST. The next two loops add in the free-stream and store angular velocities.

The remainder of the routine calculates the loading distributions and total forces and moments acting on the store. The strengths of the sources and doublets modeling the store in the nonuniform flow field are calculated by a call to SDSTRN which uses the method described in Section A-2 of Appendix A of Reference 2. Certain constants are calculated and then, in a loop over the axial stations where the loads are to be calculated, the circumferential pressure distributions are calculated and integrated to determine the loads using the methods of Section A-3. After exiting from this loop, the load distributions are integrated axially to get the forces and moments.

Subroutine references:

BVARIA, FREFSH, INTOST, NUMACH, REFSHK, RESVEL, SDSTRN, SDTRMS, SHAPE, SIMSON, STTOIN

Called by:

TRJTRY

C-71 Subroutine SFORC2

Subroutine SFORC2 organizes the calculation of the aerodynamic forces and moments on the separating elliptic store body using the three dimensional method with source panels as described in Section 6.3.1 of Reference 2. The primary functions of this routine are to determine the presence of an image store and to compute the influence of the parent aircraft components at the control points of a rotating and translating store. A listing of the routine is presented in Figure C-1(ww), and a flow chart is given in Figure C-14 of this report.

This routine is organized around computing the influence of the parent aircraft components on the store. SFORC2 first computes initial constants and sets aside array space in blank common for velocity arrays and temporary storage of the image store control points. Based on the location of the store center of gravity relative to the wing axis, the β's associated with the nonlinear shock emanating from the parent aircraft wing are computed in NUMACH. The reflections of the separating store shocks from the wing and either the circular or noncircular fuselage are then checked to see whether they strike the store in ELRFLW and ELRFLB, respectively. If they do, the closer of the two is used to locate the presence of an image store later.

SFORC2 then performs a double DO loop over the number of rings and the number of panels in a ring on the store to compute the parent aircraft influence at each panel control point. Each control point is located in the fuselage coordinate system by rotation relative to the moment center in STTOIN. The β's to be used in the velocity calculation due to nonlinear shocks

propagating from all parent aircraft axisymmetric and noncircular bodies except the separating store are computed in BVARIA. If the point lies within the influence of the wing, the wing Mach number and β are interpolated for from the values at the leading and trailing edges computed in NUMACH. The parent aircraft velocity influences are computed in RESVEL in the fuselage reference coordinate system and rotated into store coordinates in INTOST. The free stream velocity components due to the translation of the separated store previously computed in VXYZ are then added. If damping is included, it is added in VELDMP.

If the image store was to be included for the closer of the wing or the fuselage reflections, the set of control points on the image store are computed in IMAGYZ at this point. If more than one empennage is present on the store, an average of the nonuniform flow field at each ring is computed in VWAVG to be used later in tracking vortices between sets of fins. The panel strengths and the forces and moments are computed in a call to SDSTN2. This generates the loads only on the store body including the influences of the parent aircraft, the free stream, and the reflected shock in the form of an image store. All empennage effects will be included later in computing the loads on the interference shell of the empennage.

Subroutine references:

BVARIA, ELRFLB, ELRFLW, IMAGYZ, INTOST, IOREAD, NUMACH, RESVEL, SDSTN2, STTOIN, VELDMP, VNORM, VWAVG

Called by:

TRJTRY

C-72 Subroutine SHAPE

The purpose of this subroutine is to calculate the radius and surface slope for a circular body at a specified axial station. The body shape is specified by a series of polynomials of the form of Equation (1) of Volume II. A flow chart of subroutine SHAPE is presented in Figure A-11 of Volume III and a listing of the subroutine in Figure C-1(xx).

The quantities in the parameter list are:

X	value of x/ℓ at which radius and surface slope are to be calculated
NS	number of polynomials describing body shape
XE	array containing values of x/ℓ for the end points of the NS polynomials
C	array containing the coefficients of the NS polynomials
R	calculated value of r/ℓ at x/ℓ = X
DRDX	calculated value of dr/dx at x/ℓ = X

The calculation performed by this subroutine consists of two steps. The first step is to determine which of the NS polynomials describes the shape at the value of X where the radius and surface slope are required. Once this is determined, the appropriate set of coefficients is used in Equation (1) of Reference 1 to determine r/ℓ. The value of dr/dx is found by differentiating Equation (1).

$$\frac{dr}{dx} = \frac{C_7}{2}\left[\frac{2C_2\frac{x}{\ell} + C_3}{\sqrt{C_2\left(\frac{x}{\ell}\right)^2 + C_3\frac{x}{\ell} + C_4}}\right] + C_5 + 2C_6\frac{x}{\ell}$$

It should be noted that r/ℓ and dr/dx are calculated using the coefficients of the NSth polynomial if x/ℓ is greater than XE(NS).

Called by:

EJECTR, SFORCE, SWINT, TRJTRY

C-73 Subroutine SHKLOC

Subroutine SHKLOC locates the shock wave from an axisymmetric body at coordinates Y,Z relative to the body. The location is converted into a radial distance and a search performed in the tabulated shock shape to find the two values between which the point lies. A linear interpolation is then performed to compute the X location of the shock. A listing of the routine is presented in Figure C-1(yy) of this report. The descriptions of the parameters in the argument list follow:

X	computed axial location of shock at distance R from body
R	radial distance from body at which axial location is to be computed
NS	number of X versus R pairs in shock table
SX,SR	arrays containing X and R values describing shock shape relative to axisymmetric body. Shape is generated at zero degrees angle of attack and is invariant with polar angle

Called by:

BVARIA

C-74 Subroutine SIMSON

Subroutine SIMSON calculates the value of a definite integral using Simpson's rule. This can be found in any elementary numerical analysis book, for example, Reference 7. As programmed here

$$I = \int_{x_o}^{x_o+m\Delta x} f(x)\,dx \cong \frac{\Delta x}{3}\{f(x_o) + 4f(x_o + \Delta x) + 2f(x_o + 2\Delta x)$$

$$+ 4f(x_o + 3\Delta x) + 2f(x_o + 4\Delta x) + \ldots + 4f[x_o + (m-1)\Delta x]$$

$$+ f(x_o + m\Delta x)\}$$

where m must be an even number and 4 or greater. The subroutine is listed in Figure C-1(yy) of this report. Referring to the listing and the above equation, the quantities in the subroutine parameter list are:

N	$m + 1$
F	$f(x)$
DX	Δx
SUM	I

C-75 Subroutine SMARCH

Subroutine SMARCH solves for the source panel strengths using a ring by ring marching technique. The solution may be computed also for the panel strengths in the presence of an image body. A listing of the subroutine is presented in Figure C-1(yy) of this report.

SMARCH is organized to provide the general source panel solution for a body in supersonic flow:

$$[A]\gamma_B = V_B - [A_I]\gamma_B$$

where [A] is the aerodynamic influence coefficient matrix partitioned into blocks of the coefficients of one ring on another; γ_B are the panel strengths; V_B are the boundary condition normal velocities in the absence of an image body; and $[A_I]$ is the influence coefficient matrix of the image body on the real body control points. The solution proceeds in blocks of equations, with only those blocks of equations on or below the diagonal computed and saved on TAPE9. The first block corresponding to the influence of a ring on itself is read from TAPE9 in [L*U] decomposed form. The solution for that ring is computed with PAS002. Subsequent blocks in column form are read and multiplied by the strengths of that ring and subtracted from the boundary conditions for following rings. If an image body is present the influence of that ring of the image body is also computed on following downstream panels in IMAGEV and used to modify subsequent boundary conditions.

The descriptions of the parameters in the argument list follow:

GB	array of panel strengths, γ_B
VB	array of panel boundary conditions destroyed during solution
IA	index of starting location in blank common of temporary A matrix
IMAGE	logical indicator of presence of image body

Subroutine references:
IOREAD, PAS002, IMAGEV

Called by:
SDSTN2

C-76 Subroutine SOLVUV

Subroutine SOLVUV computes the panel on panel induced velocity components at panel control points. Velocity coefficient arrays are read from TAPE8 and multiplied by the panel strength to compute U,V,W.

$$U = [A_u]\gamma_B$$
$$V = [A_v]\gamma_B$$
$$W = [A_w]\gamma_B$$

The coefficient arrays are assumed to be stored in blocks of coefficients representing the influence of one ring of panels on another. The routine performs a double DO loop over the number of rings. The outer loop steps through the number of influencing blocks of data. The inner loop in supersonic flow steps through the number of rings of panels influenced. For a given ring, it is assumed there is no upstream influence and the latter loop starts from the ring on itself. A listing of the routine is presented in Figure C-1(yy) of this report. The descriptions of the parameters in the argument list follow:

U,V,W	arrays of computed velocity components in source panel coordinate system
A	temporary matrix to contain blocks of coefficients read from TAPE8
GB	array of panel strengths
NBODY	number of panels

SUPERS logical indicator representing supersonic flow
SUPERS=.true., only lower triangle blocks computed
SUPERS=.false., entire matrix of blocks used in computation

Subroutine references:
IOREAD

Called by:
SDSTN2

C-77 Subroutine SORPAN

Subroutine SORPAN computes the three components of velocity induced at a specified control point by a constant source distribution on a quadrilateral panel having longitudinal taper and inclined at an angle, DELTA, to the free stream direction. This version has been specialized for only supersonic flow. This routine is based on the methods and equations presented in Reference 8. A listing of the subroutine is presented in Figure C-1(zz) of this report.

The descriptions of the parameters in the argument list follow:

UPM,VPM,WPM three orthogonal components of velocity in local panel coordinates induced by panel with control point XJ,ZJ at field point XI,YI,ZI

Called by:
PANVEL

C-78 Subroutine SOUTPT

Subroutine SOUTPT prints the output at the end of each trajectory integration step. A listing of the subroutine is presented in Figure C-1(aaa) and a flow chart in Figure C-15 of this report.

The current value of the time is first printed and then the force and moment components calculated in subroutines SFORCE and SEMFOR for the circular store option or in subroutines SFORC2 and DEMON2 for the elliptic store option and the totals are printed. If the ejector force option is used (NJECTR not equal to zero) the ejector forces and moments are printed. If store thrust has been calculated (NTRHUS not equal to zero) the thrust force is printed. Next, the normal-force and side-force distributions acting on the store body alone are printed. The x_s locations are the points on the body at which the forces act.

The next section of the subroutine locates the store nose, moment center, and base in the fuselage or inertial system. These points are located relative to the fuselage nose and also relative to where they would be had the store remained in the t = 0 position on the aircraft. These positions are printed.

The remainder of the subroutine prints the store moment center translational velocities and accelerations, the store rotational velocities and accelerations, and the store angular orientation and rates of change of these angles.

Subroutine references:

STTOIN

Called by:

TRJTRY

C-79 Subroutine SPECPR

Subroutine SPECPR computes the Bernoulli pressures at control points of the constant u-velocity panels on the empennage fin surfaces. It calls BDYPR to compute pressures on the body interference shell and LOADS to sum the forces and moments on the empennage. A description of the methods and equations used is given in Section 3.6 of Reference 5. A listing of the routine is presented in Figure C-1(bbb) of this report.

In calculating the panel pressures only the influence of the store singularities and the store motion are accounted for. All velocities and loads computed are in the body coordinate system. The descriptions of the parameters in the argument list follow:

NDAMP	damping indicator; see input item 4 to Program II
XM	moment center relative to store nose, feet
VSTORE	V_{∞_s}, Equation (97), Reference 2
VAR(N), N=1,2,...12	$\dot{\xi}$, $\dot{\eta}$, $\dot{\zeta}$, p, q, r, ξ, η, ζ, Ψ, Θ, Φ respectively
HEAD	alphanumeric description of empennage

Subroutine references:

BDYPR, LOADS, ROTFW, ROTFW, VELDMP, VELNOR

Called by:

DEMON2

C-80 Subroutine SPNLD

Subroutine SPNLD computes the span load distribution for monoplane or cruciform wing or fin configurations. The cruciform configuration is restricted to the "plus" fin arrangement. Fin trailing edge vortex strengths and spanwise locations are determined from the span load distributions. A description of the relevant methods and equations is given in Appendix B of Reference 5. A listing of the routine is presented in Figure C-1(ccc) of this report.

Called by:
LOADS

C-81 Subroutine STRDAT

Subroutine STRDAT copies the separating elliptic store (NEJSTR) data into labeled and blank commons and organizes the remaining elliptic store and noncircular fuselage data sequentially in blank common. It sets up and saves the information necessary to locate all arrays for two elliptic store shapes and the noncircular fuselage in blank common. A listing of the routine is presented in Figure C-1(fff) of this report. A description of the locations of variables in blank common is presented in Appendix D.

This routine makes a series of calls to FRSTRT to reorganize existing data. It assumes that the noncircular fuselage data arrays have been copied by RDFILE onto TAPE11 and the similar arrays for the elliptic store shapes have been copied onto TAPE10. The data for the separating elliptic store shape is first copied by FRSTRT using KODE=4. All control integers and parameters are copied into their appropriate labeled commons. The geometric arrays are stored in blank common starting at the first location. This is the only configuration for which the aerodynamic influence

coefficients are copied onto TAPE8 and TAPE9. Additional storage space in blank common is also allocated to hold the arrays VB, U, V, W, and CP. The parameter LASTEJ is defined to contain the last location in blank common used by the separating store data. If the second elliptic shape is the separating store, the first store shape data on TAPE 10 must be read and then replaced by data for the second shape.

The data of the noncircular fuselage is copied into blank common by FRSTRT using KODE=5. For the fuselage and the additional store shape both the control integers and geometric arrays must be stored in blank common. The fuselage data is saved immediately after the separating store arrays. The parameters IDF and LASTF are defined to contain the offset value and the last location for the fuselage data in blank common.

The data of the second elliptic store shape is next copied into blank common by FRSTRT. If it is the second store shape on TAPE10, it is read using KODE=5. If it is the first shape, TAPE10 is rewound and the data read using KODE=6. The parameters ID2 and LASTA are defined to contain the offset value and the last location for the second store shape in blank common.

A copy of blank common containing the separating and second store data and the fuselage data is lastly saved on TAPE7. This is the copy of blank common that will be retrieved each time blank common is overwritten by other routines. A description of the allocation of the actual arrays in blank common at this time is found in Section D-3 of Appendix D. The descriptions of the parameters in the argument list follows:

NFU fuselage indicator; see item 4 of Program I

REFAEJ reference area of separating store, feet

REFDEJ reference length of separating store, feet

BODLEJ body length of separating store, feet

Subroutine references:

FRSTRT, IOWRIT

Called by:

TRJTRY

C-82 Subroutine STTOIN

Subroutine STTOIN (see Figure C-1(fff) for a listing) takes a vector with components specified in the store x,y,z coordinate system directions and transforms it into a vector with components in the inertial ξ,η,ζ coordinate system directions, see Figure 21 of Volume II. That is,

$$\begin{pmatrix} s_\xi \\ s_\eta \\ s_\zeta \end{pmatrix} = [A] \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix}$$

The matrix [A] is given by Equation (B-2) in Appendix B of Reference 2 and is calculated in subroutine DIRCOS for the separating store. The similar arrays for the fixed store are calculated in RESVEL.

In terms of the above notation, the quantities in the parameter list of the subroutine are:

X	s_x
Y	s_y
Z	s_z
XI	s_ξ
ETA	s_η
ZETA	s_ζ
DC	[A]

Called by:

SFORCE, SEMFOR, SFORC2, DEMON2, SOUTPT, REFSHK, ELRFLB, EJECTR, RESVEL

C-83 Subroutine SWINT

Subroutine SWINT finds the intersection of a shock with an axisymmetric body. It is used to locate the intersection of the store shock with the circular fuselage. The exact location is computed by interpolation in the tabulated store shock wave shape and an iteration with the polynomial description of the surface shape. A listing of the routine is presented in Figure C-1(fff) of this report.

The routine is partitioned into two tasks. The first is finding the axial intersection, XAX, of the tabulated shock shape with the axis of the body. If the body is located at a radial distance beyond the tabulated shape extrapolation using the last two points is performed. Otherwise the table is searched to find the two values of RS between which the axis lies. The return intersection code, INT, is set one of three values depending on whether the body intersection is found.

$$\text{INT} = \begin{cases} >1, & 0 < \text{XAX} < \text{BLN, intersection with body occurs} \\ 0, & 0 \geq \text{XAX or XAX} \geq \text{BLN, no intersection found} \\ -1, & \text{XAX search failed} \end{cases}$$

The second task performed by SWINT is a refinement of the intersection point from the axis to the point on the surface of the body. The procedure used here is to first back up in the tabulated shock shape to find the XS and RS values which bracket the radial distance to the body surface. A more precise location is found by continuing to halve the interval until the tolerance 0.01*RMAX is achieved. The value of INT contains the number of times the interval was halved.

The descriptions of the parameters in the argument list follow:

XZ,RZ	X,R coordinates of shock origin relative to axisymmetric fuselage nose
KX	number of XS-RS table values
XS,RS	table of X,R values of separating store shock relative to store nose
XE,C,NS	information giving polynomial description of fuselage, $r/\ell = f(x/\ell)$: end points, coefficients and number of segments
BLN	length of fuselage (ℓ)
RMAX	maximum radius of fuselage, used to determine tolerance
DRDXS	dr/dx of shock wave at intersection point
DRDXB	dr/dx of fuselage at intersection point
XB,RB	X,R of intersection point relative to fuselage nose
INT	intersection code parameter; see above descriptions

Subroutine references:

SHAPE

Called by:

FREFSH, ELRFLB

C-84 Subroutine SWINTE

Subroutine SWINTE finds the intersection of a shock with a noncircular arbitrarily paneled body. Only an approximate reflection point based on the intersection of the shock with the panel surface is computed. A listing of the routine is presented in Figure C-1(ggg) of this report.

The procedure used to approximate the shock reflection point, similar to that shown in Figure C-7, is:

1. Define the R versus X plane between the fuselage axis and the store nose.

$$RZ = \sqrt{Y_{BN}^2 + Z_{BN}^2}$$

$$\sin\theta_{BS} = Y_{BN}/RZ$$

$$\cos\theta_{BS} = Z_{BN}/RZ$$

   R is positive from fuselage centerline outward and X is positive from fuselage nose aft.

2. Find axial location, XAX of intersection of tabulated shock, XS,RS, with fuselage axis. If XAX is between the nose and BLN, the intersection is assumed to occur. The intersection code, IMFSTR, is set to one of the two return codes as follows. If IMFSTR=0, the remaining steps are ignored.

$$\text{IMFSTR} = \begin{cases} 1, & 0 < \text{XAX} < \text{BLN, intersection occurs} \\ 0, & \text{no intersection occurs} \end{cases}$$

3. Refine axial and meridional locations by finding the ring of panels which contains the point XAX+β*ZPT, where ZPT is the Z-coordinate of the first panel in the ring of interest. The meridional panel is located by searching for panel, IP, with outward normal lying closest to the plane containing the point and the store axis.

4. The intersection of the shock with the panel is then computed from the line through the panel control point with the same slope as the panel and the tabulated shock values.

   Equation of line in plane of panel:

$$R = \sqrt{YPT(IP)^2 + ZPT(IP)^2} + (dr/dx)_B \cdot [XBS-XPT(IP)]$$

   Equation of shock segment:

$$R = RZ - RS_{i-1} - (dr/dx)_s (XBS-XS_{i-1}-XZ)$$

   where

$$(dr/dx)_s = -(RS_i - RS_{i-1})/(XS_i - XS_{i-1})$$

   Intersection at:

$$XBS = \frac{RZ-RS_{i-1}-(dr/dx)_s(XS_{i-1}+XZ) - \sqrt{YPT(IP)^2+ZPT(IP)^2} + (dr/dx)_B \cdot XPT(IP)}{(dr/dx)_B - (dr/dx)_s}$$

5. The intersection location, XBS,RBS, is converted into fuselage coordinates and projected onto the line between the body axis and the store nose.

Fuselage coordinates of intersection:

$$Y'_{BS} = (dr/dx)_B(X_{BS} - XPT(IP))\ \sin\theta_{IP} + YPT(IP)$$

$$Z'_{BS} = (dr/dx)_B(X_{BS} - XPT(IP))\ \cos\theta_{IP} + ZPT(IP)$$

Coordinates projected onto fuselage/store plane:

$$Y_{BS} = R'_{BS}\ \sin\theta_{BS} + Y_{BN}$$

$$Z_{BS} = R'_{BS}\ \cos\theta_{BS} + Z_{BN}$$

where

$$R'_{BS} = (Y'_{BS} - Y_{BN})\sin\theta_{BS} + (Z'_{BS} - Z_{BN})\cos\theta_{BS}$$

The descriptions of the parameters in the argument list follow. The remaining variables output from this routine may be found in the descriptions for common BSWINT in Appendix D.

XZ,YBN,ZBN	X,Y,Z coordinates of shock origin relative to fuselage body nose
NS	number of XS-RS table values
XS,RS	table of X,R values locating shock relative to shock origin in plane between fuselage centerline and store nose
BLN	length of body

XBS,RBS	X,R of intersection point relative to fuselage nose
NFUS	number of fuselage body segments
IMFSTR	intersection return code; see above
KFORX,KRADX	arrays of number of axial and meridional points used to describe fuselage panel geometry per segment
XPT,YPT,ZPT	arrays containing coordinates of fuselage panel control points
THET,DELTA	arrays containing inclination and incidence angle of fuselage panels at control points
XC	array of axial locations of starting and ending stations of panel rings

Called by:

FRESHK, ELRFLB

C-85 Subroutine THRCAL

Subroutine THRCAL calculates the store thrust at a given time. A listing of the subroutine is presented in Figure C-1(hhh) of this report.

The thrust force acts along the store longitudinal axis and is specified by a series of polynomials of the form

$$F_T = \sum_{n=1}^{6} a_n t^{n-1}$$

where F_T is the thrust in pounds at time t. The time history is specified by one of NTPOLY polynomials each of which is applicable for a range of t. The subroutine first determines which polynomial should be used for the given time value t. Once this is determined, the appropriate set of coefficients, a_n, is used in the above equation to calculate F_T. If t is greater than the end of the specified thrust time history, an error message is printed (see Section 4.4 of Volume II) and the program stops.

The following table of definitions contains most of the variable names used in the subroutine. Section 4.2.2 of Volume II should be consulted for the definition of a variable which is an input item.

FTHRUS	F_T; store thrust at time t
NTPOLY	number of polynomials; input item 29 of Program II
T	t; time at which thrust force is to be calculated
TC(J,I)	thrust polynomial coefficients; input item 31 of Program II
TEND(J)	time end point for Jth polynomial; input item 30 of Program II

Called by:
TRJTRY

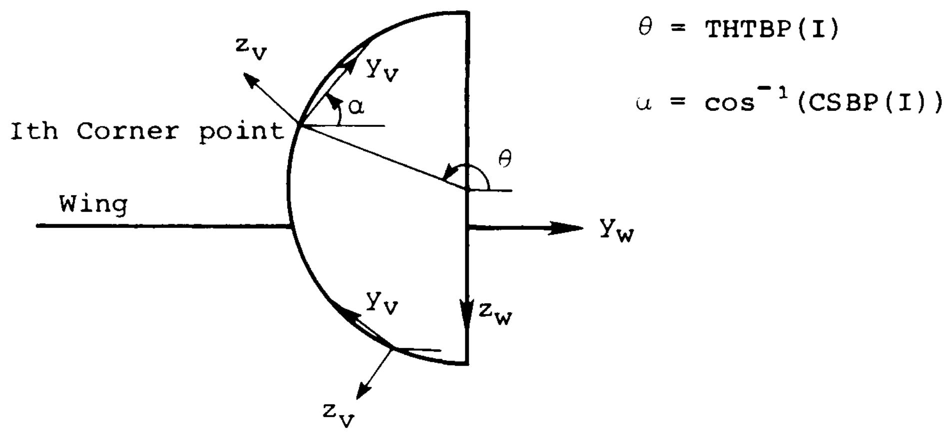
C-86 Subroutine VELBD2

Subroutine VELBD2 calculates perturbation velocities at a given field point due to the constant u-velocity panels on the

fuselage, according to the methods described in Sections 5.6 and 3.3 of Reference 2. A listing of the subroutine is presented in Figure C-1(hhh). The subroutine uses quantities calculated in subroutine BLYOUT which is described in Section A-5 of Volume III. The wing coordinate system is shown in Figure 7 of Volume II.

At the beginning of the subroutine the logical variable PYPNL is set equal to FALSE, the velocity totals UP, VP, and WP are initialized to zero, and the panel leading-edge and trailing-edge slopes, EM, are defined as zero. The remainder of the subroutine consists of a double DO loop in which the aerodynamic influence coefficients and induced velocities are calculated for a single panel corner. The outer loop index runs over the corner points in a ring on the body. The inner loop index runs over the points in a chordwise row. Variables which are constant for a chordwise row are initialized in the outer loop. The numbering of fuselage panel corners is illustrated in the sketch in Section A-29 of Volume III.

The coordinates of the field point are given as formal parameters in the wing coordinate system. The subroutine locates the field point relative to the influencing corner in the coordinate system associated with subroutine VELO2 and illustrated in Figure 7 of Reference 2. In the VELO2 system the origin is at the vertex of the semi-infinite triangle (the corner point), x_V is positive to the rear, y_V is positive to the right in the plane of the fuselage panel, and z_V is positive outward normal to the panel. The sketch below illustrates the x_V, y_V, z_V system for two panel corners, one in the upper left quadrant ($90^\circ < $THTBP(I)$ < 180^\circ$) and one in the lower left quadrant, when viewed from the rear.



At the beginning of the inner DO loop two tests are performed to eliminate possible unnecessary calculations: first, if the field point is located ahead of the corner (x<0), the influence of that corner and all of the remaining corners in the chordwise row are zero; second, if the net strength associated with the corner is zero, the perturbation velocity is zero and calculation of the influence coefficients is omitted.

The remainder of the inner loop calculates the corner influence functions, by means of subroutine VELO2, and the influence functions for the image of the corner with respect to the vertical plane of symmetry. Separate sections of the subroutine are used depending upon the fuselage quadrant in which the corner lies. The returned functions, U,V,W, are resolved back into the wing system and summed. If the corner is located in the lower left quadrant the signs of the influence functions are reversed in the summation. The flow of the logic in this routine for the calculations in the upper and lower left quadrants follows closely that in VELBD1 in Section A-47 of Volume III for corner 1 and its corresponding flow chart in Figure A-14. Only the additional loops over the corners to sum the above mentioned results have been added.

Finally, for each corner whose influence at the field point is nonzero (FELT=TRUE), the influence coefficients are multiplied by the net corner strength to obtain the induced perturbation velocities, UP,VP,WP. These velocities are summed for all corner points associated with the fuselage constant u-velocity panels.

The variables in the subroutine parameter list are:

XX,YY,ZZ	x_w, y_w, z_w coordinates of field point in wing coordinate system

Subroutine references:

VELO2

Called by:

RESVEL

C-87 Subroutine VELCAL

Subroutine VELCAL calculates perturbation velocities at a given field point due to the circular fuselage, rack or store source and doublet distributions, according to Equation (16) of Reference 2. A listing of the subroutine is presented in Figure C-1(iii) of this report. The fuselage coordinate system is shown in Figure 5 of volume II. The rack and store coordinate systems are similarly oriented.

The coordinates of the field point are given as formal parameters in the appropriate body coordinate system. The subroutine first transforms these coordinates into the VELCAL system by changing the sign of X and Z, and then into the polar coordinates XFIELD, RFIELD, and THETA.

The major part of the program consists of a DO loop within which the axial, radial, and tangential velocities due to the N sources and doublets are calculated and summed. A test is made to determine whether the field point is ahead of the Mach cone

from the Ith source origin, TX(I). If so, all remaining perturbation velocities are equal to zero and no further calculations are performed within the loop. At the end of the subroutine, the velocities are resolved back into the directions of the body coordinate system.

The variables in the subroutine parameter list are:

T	array containing the source strengths
TC	array containing the doublet strengths
TX	array containing the x locations of the origins of the singularities; positive, measured aft from tip of nose
N	number of line sources and doublets
XP	x coordinate of field point in body system
Y	y coordinate of field point in body system
ZP	z coordinate of field point in body system
U1	u/V_∞ perturbation velocity at field point; body system
V1	v/V_∞ perturbation velocity at field point; body system
W1	w/V_∞ perturbation velocity at field point; body system
BETAL	value of β used in the velocity calculation
BODYL	length of body
SUMK	sum of the source strengths
SUMKD	sum of the doublet strengths

Called by:
RESVEL

C-88 Subroutine VELDMP

Subroutine VELDMP adds to the velocities at a point the contributions due to the angular motion of the store. This routine is called only if NDAMP≠0. For each control point the moment arm

relative to the moment center is computed and the velocities due to p, q, and r are computed as follows.

$$u_i = -\frac{q}{V_{\infty_s}} ZPT_i - \frac{r}{V_{\infty_s}} YPT_i$$

$$v_i = \frac{r}{V_{\infty_s}} (XPT_i - XMOM) - \frac{p}{V_{\infty_s}} ZPT_i$$

$$w_i = \frac{q}{V_{\infty_s}} (XPT_i - XMOM) + \frac{p}{V_{\infty_s}} YPT_i$$

A listing of the routine is presented in Figure C-1(jjj) of this report. The descriptions of the parameters in the argument list follow:

UT,VT,WT	arrays containing the velocities to which the components due to rotational motion are added
XPT,YPT,ZPT	arrays containing the control points at which the velocities are computed (store body or empennage coordinates)
NCOMPT	number of control points
XMOM	location of moment center in coordinate system of control points
VSTORE	V_{∞_s}, store free-stream velocity
VAR(N) N=1,2,...12	$\dot{\xi}$, $\dot{\eta}$, $\dot{\zeta}$, p, q, r, ξ, η, ζ, Ψ, Θ, Φ respectively

Called by:

SFORC2, DEMON2, SPECPR, BDYPR

C-89 Subroutine VELNOR

Subroutine VELNOR calculates the perturbation velocities induced by the empennage fin panels and body interference panels using the superposition scheme for the influence of four corners (see Appendix A of Reference 5). VELNOR returns the velocities UP,VP,WP in empennage reference coordinate system. A description of the methods and equations used in this routine are found in Appendix A of Reference 5. A listing of the routine is presented in Figure C-1(jjj) of this report. The description of the parameters in the argument list follows:

XX,YY,ZZ	coordinates of the control point at which the influence of panels II through IF are to be summed. Point is in empennage reference axes.

Subroutine references:

ROTBW, ROTFW, ROTWB, ROTWF, VELO

Called by:

CRFWBD, SPECPR, BDYPR

C-90 Subroutine VELO

Subroutine VELO calculates the influence of the basic semi-infinite triangles which are under constant loading. It is computed under the assumption of constant U difference, u_+/V_{∞_s}. The coordinate system used here is the coordinate system associated with the triangle under consideration. The descriptions and equations used here are essentially the same as those used in VELO2 and described in Appendix II of Reference 9. A listing of the routine is presented in Figure C-1(lll) of this report.

Called by:

VELNOR

C-91 Subroutine VELO2

Subroutine VELO2 calculates the aerodynamic influence functions of a semi-infinite triangle associated with a constant u-velocity panel, as described in Section II-2.1 of Appendix II of Reference 9. The influence functions relate the panel singularity strength to the perturbation velocities induced by the triangle at a given point. They occur as the coefficients of $1/\pi(u_+/V_\infty)$ in Equations (II-4) and (II-12) of Reference 9. A listing of the subroutine is presented in Figure C-1(mmm) and a flow chart of the equivalent routine in Program I in Figure A-15 in Volume III. The coordinate system used by the subroutine is shown in Figure 3 of Reference 9.

At the beginning of the subroutine the quantity BETA is set equal to BETANU or to BETAOL depending on the values of INUMCH and PYPNL (Section C-52 of this report should be referred to for further details concerning this choice). Next, the logical variable FELT is initialized to TRUE and a test is performed to determine if the field point is located ahead of the influencing triangle ($X \leq 0$). If so, the influence functions U,V,W are set to zero, FELT is set to FALSE, and control is returned to the calling program.

Next, the variable PYPNL is tested and, if the triangle is on the pylon, a transformation is performed which rotates the triangle into the VELO2 x,y plane. After the calculation of the logical variable INSIDE and other frequently used quantities, the remainder of the subroutine consists of four major sections in which the influence function terms, F1, F2, F4, F5, and F7, are calculated. Each section corresponds to a condition of the slope, EML, of the

leading edge associated with the semi-infinite triangle. The subroutine requires that EML $\geq$ 0 and this is accounted for in the VELO2 calling programs. The four leading-edge conditions are described fully, with accompanying sketches in Section II-2.1 of Reference 9. All equation numbers mentioned in the following paragraphs are from Section II-2.1 of Reference 9.

The first section of the subroutine corresponds to a subsonic leading edge, BTSQ < EMLSQ. Equation (II-5) is used if the point is inside the Mach cone from the origin, INSIDE=TRUE. If not, U,V, and W are set to zero. In this section as in the remaining ones, discontinuities in some of the equations may occur for certain field point locations. In such cases the affected influence function is set to zero. The quantities YYEDGE and TLRNC, as well as Y and Z, are used to test the singularity locations.

If BETASQ=EMLSQ, the leading edge is a sonic leading edge. The equations used are the same as for the subsonic case except for the function F2, which is given by Equation (II-7). If the point lies outside the Mach cone from the origin, the influence functions equal zero.

The third section of the subroutine is used if the triangle leading edge is supersonic, BTSQ > EMLSQ. Equations (II-5) and (II-9) calculate the terms of the influence functions if INSIDE=TRUE. If not, a second test is performed and Equation (II-11) is used if the point is inside the Mach cone whose origin is on the leading edge at the field point y location, otherwise the functions U,V,W are set to zero.

The fourth section of the subroutine is executed if the leading edge is unswept, EML=0. For this special case the perturbation velocity equations are given by (II-12). If

INSIDE=TRUE, the influence function terms are given by (II-5) and (II-13). For points outside the Mach cone from the origin but inside the cone from the leading edge Equation (II-14) is used; otherwise, U,V,W are set to zero.

The last part of the subroutine calculates the functions U,V,W from the component terms, in the case of a leading edge with positive sweep, using Equation (II-4). If the triangle is located on the pylon, V and W are rotated back into pylon orientation.

Called by:
    VELBD2, VELPP2, VELWP2

C-92 Subroutine VELOT2

Subroutine VELOT2 calculates the aerodynamic influence functions of a semi-infinite triangle associated with a wing or pylon thickness panel, as described in Section II-2.2 of Appendix II of Reference 9. The influence functions relate the panel source strength to the perturbation velocities induced by the triangle at a given point. They occur as the coefficients of $1/\pi(\tan\theta)$ in Equations (II-15) and (II-16) of Reference 9. A listing of the subroutine is presented in Figure C-1(ooo) of this report. The subroutine is very similar in logic to subroutine VELOT1 which is described in detail in Section A-51 and represented by a flow chart in Figure A-15 in Volume III. Additional logic has been incorporated to use β_{OL} for pylons and the wing shock value, β_{ν}, for wing thickness panels.

In subroutine VELOT2, three component terms, F1, F2, and F5, need to be calculated in order to determine the influence functions UTH, VTH, and WTH. Referring to Sections II-2.1 and II-2.2 of Reference 9, function F1, F2, and F5 are specified in Equation (II-5) for the case of a subsonic leading edge, in Equations

(II-5) and (II-7) for a sonic leading edge, and in Equations (II-5), (II-9), and (II-11) for a supersonic leading edge. For the special case of an unswept leading edge (EML=0), the general perturbation velocity equations are given by (II-16). If the given point lies inside the Mach cone from the origin of the triangle, function F1 is given by Equation (II-13), function F5 by Equation (II-5), and function F2 by Equation (II-17). If the point lies outside the Mach cone from the origin but inside the Mach cone from the triangle leading edge at the field point y location, functions F1, F2, and F5 are given by Equation (II-16). In all leading edge cases, the function F1, F2, and F5 are singular for certain field point locations. When this occurs, the affected influence function is set to zero. All influence functions are set equal to zero if the point lies in the plane of the semi-infinite triangle (ZTH=0).

A table equating the algebraic and program notation for variables in common for subroutine VELOT2 is presented in Appendix D-2 of this report. One should note that the influence functions, U,V,W and the point coordinates, YS,ZS, in common VELARG are named UTH, VTH, WTH, YTH, and ZTH, respectively, in subroutine VELOT2.

Called by:

VELWT2, VELPT2

C-93 Subroutine VELPP2

Subroutine VELPP2 calculates perturbation velocities at a field point due to the constant u-velocity panels on the pylon according to methods described in Sections 5.6 and 3.3 of Reference 2. A listing of the subroutine is presented in Figure C-1(ppp). The wing coordinate system is shown in Figure 7 of Volume II.

At the beginning of the subroutine, the logical variable PYPNL is set equal to TRUE, indicating to subroutine VELO2 that calculations are to be performed for a pylon panel. The quantities YDIR and YIMG are calculated and the velocity totals, UP,VP,WP are initialized to zero. The remainder of the subroutine consists of a double DO loop in which the aerodynamic influence coefficients and induced velocities at the specified field point are calculated for a single panel corner. The outer loop index runs over the corner points in a spanwise row, the inner loop index over the points in a chordwise row. The remaining flow of calculations in this routine follow closely those in VELPP1 in Section A-52 of Volume III for corner 1 and its corresponding flow chart in Figure A-16. Only the additional loops over the corners mentioned above to sum results have been added.

The coordinates of the field point are given as formal parameters in the wing coordinate system. The subroutine locates the field point relative to the influencing corner in the coordinate system associated with subroutine VELO2 and illustrated in Figure 7 of Reference 2. At the beginning of the inner loop two tests are performed to eliminate possible unnecessary calculations: first, if the field point is located ahead of the corner, the influence of that corner and all of the remaining corners in the chordwise row are zero; second, if the net strength associated with the corner is zero, the perturbation velocity is zero and calculation of the influence coefficients is omitted.

The remainder of the inner loop calculates the corner influence functions by means of subroutine VELO2. Unless the pylon is located under the fuselage centerline (CENTER=TRUE), the influence of the image of the corner with respect to the aircraft vertical plane of symmetry is also calculated. Separate sections of the subroutine performs these calculations depending on the sign of the

leading-edge slope [illegible] the corner. The separate [illegible] the final coordinate transf[illegible] upon this leading edge sl[illegible] resolved back into the win[illegible] leading edge is swept forward, [illegible] rence functions are reversed in the [illegible].

Finally, for each corner wh[illegible] is nonzero (FELT=TRUE), the [illegible] by the net corner strength [illegible], UP,VP,WP. These velocities are [illegible] associated with the pylon constant [illegible].

A description of the parameters in the [illegible].

XX,YY,ZZ coordinates of the field point [illegible] reference axes at which the [illegible] are computed

Subroutine references:
VELO2

Called by:
RESVEL

C-94 Subroutine VELPT2

Subroutine VELPT2 calculates perturbation velocities at a field point due to the pylon thickness distribution according to methods described in Section 3.3 of Reference 2. The program logic is very similar to that of subroutine VELPP2 which calculates velocities induced by the pylon constant u-velocity panels. Subroutine VELPP2 has been described in detail in Section C-93.

only those details, therefore, in which subroutine VELPT. differs from subroutine VELPP. are included in this description. A listing of the subroutine is presented in Figure C-1(qqq) of this report.

The coordinates of the field point are given as formal parameters in the wing coordinate system. The point is located relative to each of the panel corners using the same transformation scheme as in VELPP.. However, the corner coordinate arrays which define the pylon source panel corners are used in the transformations. Subroutine VELCT. is called to calculate the corner influence functions, U,V,W which are then superposed in the same manner as in VELPP..

Finally, the influence coefficients for the Ith corner are multiplied by the net strength associated with the corner, THTNET(I), to obtain perturbation velocities induced at the field point. These velocities are calculated and summed for all corner points associated with the pylon source panels.

A description of the parameters in the argument list follow.

XX,YY,ZZ   coordinates of the field point in the wing reference axes at which the pylon panel influences are computed

Subroutine references:

VELCT.

Called by:

RESVEL

C-95 Subroutine VELWP2

Subroutine VELWP2 calculates perturbation velocities at a field point due to the wing constant u-velocity panels according to methods described in Sections 5.6 and 3.3 of Reference 2. The program logic is very similar to that of subroutine VELPP2 which calculates velocities induced by the pylon constant u-velocity panels. Subroutine VELPP2 has been described in detail in Section C-93. Only those details, therefore, in which subroutine VELWP2 differs from subroutine VELPP2 are included in this description. A listing of the subroutine is presented in Figure C-1(rrr) of this report.

At the beginning of the subroutine the logical variable PYPNL is set equal to FALSE, indicating to subroutine VELO2 that calculations are to be performed for a wing rather than a pylon panel. The coordinates of the field points are given as formal parameters in the wing coordinate system. The point is located relative to the influencing corner in the VELO2 coordinate system using the same transformation scheme as in VELPP2. However, wing dihedral effects, if nonzero, are accounted for in the transformations. Variables ZDIHED, CPHI, and SPHI are used for this purpose. Finally, since the image panel corner is always present on the right wing panel, no test for symmetry (as in the case of a centered pylon) is necessary in this subroutine. Velocities are calculated and summed for all corner points associated with the wing constant u-velocity panels.

The description of the parameters in the argument list follows:

XX,YY,ZZ	coordinates of the field point in the wing reference axes at which the wing panel influences are computed

Subroutine references:
VEL02

Called by:
RESVEL

C-96 Subroutine VELWT2

Subroutine VELWT2 calculates perturbation velocities at a given field point due to the wing thickness distribution, according to methods described in Section 3.3 of Reference 2. The program logic is very similar to that of the corresponding subroutine VELPP2, which calculates velocities induced by the pylon constant u-velocity panels. Subroutine VELPP2 has been described in detail in Section C-93. Only those details, therefore, in which subroutine VELWT2 differs from subroutine VELPP2 are included in this description. A listing of the subroutine is presented in Figure C-1(sss) of this report.

At the beginning of the subroutine the logical variable PYPNL is set equal to FALSE, indicating to subroutine VELOT2 that calculations are to be performed for a wing rather than a pylon panel. The coordinates of the field point are given as formal parameters in the wing coordinate system. The point is located relative to the influencing corner in the VELOT2 coordinate system using the same transformation scheme as in VELPP2. However, the corner coordinate arrays which define the wing source panel corners are used in the transformations. Also, wing dihedral effects, if nonzero are accounted for. Variables ZDEHED, CPHS, and SPHS are used for this purpose. Subroutine VELOT2 is called to calculate the corner influence functions. Since the image panel corner is always present on the right wing panel, no test for symmetry (as in the case of a centered pylon) is necessary in this subroutine.

Finally, the influence coefficients for the Ith corner are multiplied by the net strength associated with that corner, THTNET(I), to obtain perturbation velocities induced at the field point. These velocities are calculated and summed for all corner points associated with the wing source panels.

The description of the parameters in the argument list follows:

XX,YY,ZZ — coordinates of the field point in the wing reference axes at which the wing panel influences are computed

Subroutine references:
VELOT2

Called by:
RFSVEL

C-97 Function VNORM

Function VNORM computes the velocity normal to the surface of the Ith source panel. The panel is assumed to have velocities U,V,W at the control point in the source panel reference coordinate system in Figure 13 of Volume II. The panel is oriented at angles θ and δ relative to the reference axes. A listing of the routine is presented in Figure C-1(ttt) of this report.

The descriptions of the parameters in the argument list follow:

I — panel index of source panel

U,V,W — arrays containing the orthogonal velocity components in the source panel reference coordinate system

THET — array containing the source panel inclination angles, θ

DELTA — array containing the source panel incidence angles, δ

C-98 Subroutine VOTEX

Subroutine VOTEX computes the perturbation velocity components at the vortex locations accounting for mutual interference effects and the presence of an elliptical cross section. A description of the methods and equations used in this routine are found in Section 5.1 of Appendix I of Reference 5. A listing of the routine is presented in Figure C-1(ttt) of this report. The descriptions of the parameters in the argument list follow:

NV — number of vortices

XV(I),YV(I) — y and z coordinates of the Ith vortex path

GV(I) — γ, strength of the Ith vortex

VV(I),WW(I) — velocity induced at the vortex location of the Ith vortex in the y and z coordinate directions

Subroutine references:
DSDZ, DZDS, D2SDZ2, Z

Called by:
F

C-99 Subroutine VPATH

Subroutine VPATH organizes the data transfer from TRJTRY to VPATHL when vortices are present with multi-empennage configurations. A listing of the routine is presented in Figure C-1(ttt) of this report. The descriptions of the parameters in the argument list follow:

NOUTG	print control index; 0=no, 1=yes
NVRTX	number of vortices
VRTMAX	maximum vorticity; see input item 23 to Program II

Subroutine references:
VPATHL

Called by:
TRJTRY

C-100 Subroutine VPATHL

Subroutine VPATHL sets up and computes the paths and vortex induced crossflow velocities at specified field points for a set of vortices in the presence of a body in a nonuniform flow field. Slender body theory is used in the computation of crossflow velocities. The effect of the nonuniform flow field is incorporated by averaging the velocities around a ring of panels on the body to a single value at the centerline. The present program is an adaptation of the methods and equations described in Appendix I of Reference 5. The routine is specialized for elliptic bodies only and adapted for nonuniform flow fields. The coordinate system used here is the body coordinate system with the x-axis

along the body centerline starting at the nose tip, y-axis to the right when looking forward, and z-axis up. A listing of the routine is presented in Figure C-1(uuu) of this report. The descriptions of the parameters in the argument list follow:

XIP	array of x-stations used for integration steps
NVV	number of vortices
NIP	number of x-stations between which path integration is computed
E5	integration tolerance; set E5=0.001
VRTMAX	maximum vorticity; see input item 23 to Program II
LPRT	logical print control; .T.=yes, .F.=no
XCP,YCP,ZCP	field points at which vorticity is computed (not used since NCP=0)
VCP,WCP	arrays of crossflow velocities computed at above field points (not used since NCP=0)
NCP	number of field points at which velocities are to be computed (set to zero in VPATH)

Subroutine references:
DASCRU, ELLSHP, VVELS

Called by:
TRJTRY

C-101 Subroutine VVELS

Subroutine VVELS computes the perturbation velocity components due to NV external vortices and their images inside a body with elliptic cross section. This routine is used to compute the influence of vortices with known paths and strengths at specified control points by DEMON2. The crossflow velocities, v and w, are added to the input values. The methods and equations used in this routine are described in Sections 5.1 and 5.2 of Appendix I of Reference 5. A listing of the routine is presented in Figure C-1(vvv) of this report. A description of the parameters in the argument list follow.

NV	number of vortices
YY,ZZ	y and z coordinates in crossflow plane of control point
VX(I),VY(I)	y and z coordinates of path of Ith vortex at x-station
G(I)	γ, strength of Ith vortex
AB,BB	vertical and horizontal semi-axes of elliptic body cross section
V,W	crossflow velocities, v and w, at control point with vortex contribution added
VRTMAX	maximum vorticity; see input item 23 to Program II

Subroutine references:

DSDZ, Z

Called by:

DEMON2, VPATHL

C-102 Subroutine VWAVG

Subroutine VWAVG computes the average of the externally applied velocities for all the source panels in a given ring of panels on the store body. These velocity properties are used by the slender body theory calculations in VPATHL as the average nonuniform flow field seen by the external vortices. For a given ring the V,W and X values of the control points are summed and divided by the number of panels on a ring to find the average values. The computed averages are stored in labeled common WDY1. A listing of the routine is presented in Figure C-1(vvv) of this report. A description of the parameters in the argument list follow.

V,W	arrays of externally applied velocities at source panel control points
X	array of axial stations of source panel control points
NRING	number of rings of panels
IROW(J)	number of panels in the Jth ring of panels

Called by:

SFORC2

C-103 Subroutine VXYZ

Subroutine VXYZ computes the velocity components due to the free stream from the translation of the separating store body

center of gravity. It computes three functions used by the store force calculation routines in adding the influence of the free stream. It computes the direction cosine matrix, [DC], by calling DIRCOS. It then computes the relative velocity components and rotates them into the store coordinate system through INTOST.

$$[DC] = f(\psi, \Theta, \Phi)$$

if NGAM = 1,

$$[DC] = f\left(\psi - \tan^{-1}\left(\frac{\dot{\eta}}{V_\infty \cos\alpha_{CR} + \dot{\xi}}\right),\ \Theta - \tan^{-1}\left(\frac{\dot{\zeta}}{V_\infty \cos\alpha_{CR} + \dot{\xi}}\right),\ \Phi\right)$$

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = [DC] \begin{pmatrix} V_\xi \\ V_\eta \\ V_\zeta \end{pmatrix}, \quad \text{where} \quad \begin{aligned} V_\xi &= V_\infty \cos\alpha_{CR} \\ V_\eta &= 0 \\ V_\zeta &= V_\infty \sin\alpha_{CR} \end{aligned}$$

or if NGAM = 0,

$$\begin{aligned} V_\xi &= V_\infty \cos\alpha_{CR} + \dot{\xi} \\ V_\eta &= \dot{\eta} \\ V_\zeta &= V_\infty \sin\alpha_{CR} + \dot{\zeta} \end{aligned}$$

The remaining terms computed express the above quantities in terms of angles required by various routines. A listing of the routine is presented in Figure C-1(www) of this report. A description of the remaining parameters computed are primarily found in Section D-2 of Appendix D under the definitions of labeled commons BVELFS and PARAM.

Subroutine references:

DIRCOS, INTOST

Called by:

TRJTRY

C-104 Subroutine XVSR

Subroutine XVSR performs the search of the tabulated data of a single shock shape and linearly interpolates between values to find the value of XS for the given value of RS. In addition, a rotation of the shock shape is made during the interpolation to account for angle of attack effects. A listing of the routine is presented in Figure C-1(www) of this report. The descriptions of the parameters in the argument list follow:

RSHK,XSHK	arrays containing the R and X values of the table describing a single nonlinear shock shape
NSHK	number of values in shock shape
RS	input value of R at which the value of XS is to be computed
XS	output interpolated value of X locating shock
ALP	resultant angle through which shape is rotated before interpolation

Called by:
INLBET, XVSRT

C-105 Subroutine XVSRT

Subroutine XVSRT interpolates for the X-location of the nonlinear shock shape versus the radial distance from the centerline and the meridional angle around a noncircular body. The interpolation may also include the rotation of the shock shape to account for the effects of angle of attack. A listing of the routine is presented in Figure C-1(www) of this report.

This routine makes two assumptions with regard to symmetry of the shock shape. On routine entry, the Y and Z coordinates are converted to R and meridional angle, θ, measured from the z-axis, negative down. If the parameter LSYM=.true., it is assumed that the tabulated shock shape was generated at $\alpha = 0^{\circ}$, between $0^{\circ} < \theta < 90^{\circ}$, and that both right-left and top-bottom symmetry exist. θ is then converted to the equivalent angle in the quadrant between 0° and 90°. If LSYM=.false., it is assumed that the tabulated shock shape was generated at angle of attack and the computed shape has the α effect already in it. Only right-left symmetry is presumed. θ is then converted to an equivalent angle between 0° and 180°.

Three options are available for computing the axial location of the shock shape depending on the number of shock traverses used in generating the shape. If NSHOCK$\leq$0, the linear theory Mach wave corrected for angle of attack and roll is used to compute XS. If NSHOCK=1, the shape is constant everywhere and only one traverse is interpolated in and corrected for angle of attack. If NSHOCK>1, the x-value at the two closest shock traverses are computed, and the shock location XS is computed by linear interpolation in θ.

The descriptions of the parameters in the argument list follow:

XS	computed x-value of shock shape relative to body nose
YS,ZS	y,z coordinates of point at which shock is computed in body source panel coordinates
NSHK	array of the number of points in each tabulated shock for each meridional location
PHIS	array of meridional angles of each shock traverse

NSHOCK	number of meridional locations of shock traverses
XSHK,RSHK	x and r values of tabulated shock shape for all meridional angle of traverse
ALP	angle of attack which shock must be rotated through; $\alpha = (\alpha_{CR} - \text{SIBCR})(1 - \varepsilon_{\alpha})$
PHIR	angle of roll of body
LSYM	symmetry option: .true. - right-left and top-bottom symmetry .false. - right-left symmetry only

Subroutine references:
XVSR

Called by:
BVARIA, ELRFLB, ELRFLW

C-106 Subroutine Z

Subroutine Z calculates the sigma value in the transformed (circle) plane for a given tau in the physical plane for an elliptical body with wings. The methods and equations used here are described in Section 5.1 of Appendix I of Reference 5. A listing of the routine is presented in Figure C-1(xxx) of this report.

Subroutine references:
DBLU

Called by:
PITROL, VOTEX, VVELS

C-107 Subroutine ZIMAGE

Subroutine ZIMAGE organizes the calculation of image store induced velocities at the control points on the fins of the separated store. This routine is called by SEMFOR when the circular store option is used. The equations used in calculating the source and doublet induced velocities are presented in Appendix A of Reference 2. A listing of the routine is presented in Figure C-1(xxx).

At the beginning of the routine the free-stream axial and total Mach numbers as seen by the store are calculated. The shoulder location of the separated store is defined. If a wing image store is present, IMSTOR≠0, the fin control point is located relative to the image store and other quantities used in the image store velocity field calculation are determined. If fuselage image stores are present, IMFSTR≠0, similar quantities are calculated.

The next part of the routine calculates the wing image store induced velocities at the fin control point if a wing image store is present, IMSTOR≠0. The values of β_a and β_s are determined and subroutine ZIMSVL is called to calculate the wing image store induced velocities. The velocities to be returned to SEMFOR are set equal to these velocities.

The next part of the routine repeats the above calculation for the fuselage image stores if they are present, IMFSTR≠0. The values of β_a and β_s are calculated for the image store and the centerline image store. Two calls are then made to ZIMSVL to calculate the velocities induced by these stores. The velocities due to the image store are added to the previously calculated velocities and those due to the centerline store are subtracted.

The descriptions of the parameters in the subroutine argument list follow:

XFP,YFP,ZFP   x,y,z coordinates of fin control point in separated store coordinate system; origin at store nose with x positive aft, y positive to the right, and z positive up

UIM,VIM,WIM   u,v,w velocity components at fin control point in separated store coordinate system; u positive aft, v positive to the right, and w positive up

Subroutine reference:

ZIMSVL

Called by:

SEMFOR

C-108 Subroutine ZIMSVL

Subroutine ZIMSVL calculates the velocities induced by a circular image store at a fin control point on the real circular store. A listing of the subroutine is presented in Figure C-1(xxx). The equations programmed in this routine are derived in Appendix A of Reference 2

The routine consists of two DO loops followed by a summing up of the axial, radial, and tangential velocity components and a resolution of the latter two into the real store coordinate system.

The first loop calculates the line source induced velocities using Equation (A-15) of Reference 2. The first source has its origin at the image store nose and successive sources have their origins downstream. A test in the loop is made to determine whether a source influences the field point. If it does not, a transfer out of loop takes place since the following sources also cannot influence the field point.

The second DO loop calculates the velocities induced by the two sets of line doublets using Equations (A-24) and (A-32) of Reference 2. The calculation is performed in a manner identical to that previously described for the sources.

Following the second loop the axial, radial, and tangential velocities are summed up and the latter two transformed to v and w velocities in the store coordinate system.

$$v_{SD} = -V_{RAD} \sin\phi_r + V_{TAN} \cos\phi_r$$

$$w_{SD} = -V_{RAD} \cos\phi_r - V_{TAN} \sin\phi_r$$

The descriptions of the parameters in the subroutine argument list follow:

NM	number of line sources or line doublets
XFP	axial location of field point relative to image store nose
RFP	radial location of field point relative to image store longitudinal axis

CTHET, STHET, — $\cos\theta$ and $\sin\theta$ where θ is the angle between the image store z axis and the line connecting the store axis with the field point

BETAAL — local value of β_a used in the line source calculation

BETASL — local value of β_s used in the line doublet calculation

USD,VSD,WSD — u,v, and w velocity components in the real store coordinate system induced at a control point by the image store

FAC — image store doublet multiplicative factor:

- FAC = 1.0 for wing image
- $0 \leq$ FAC ≤ 1.0 for circular fuselage image store, calculated in FREFSH
- FAC = 0 for circular fuselage centerline image store

CTHER, STHER — $\cos\theta_r$ and $\sin\theta_r$ where θ_r is the angle which resolves the image store induced velocities back into the separated store coordinate system

Subroutine reference:

SDTRMS

Called by:

ZIMAGE

Figure C-1.- Listing of Program II

(Pages 155 through 231)

```
      PROGRAM TRJTRY(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE12
     1   ,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE3)
C
C     PROGRAM TO CALCULATE SIX-DEGREE-OF-FREEDOM STORE TRAJECTORIES
C          AT SUPERSONIC SPEEDS
C
C     THIS IS PROGRAM 2
C
C     THIS PROGRAM READS IN THE DATA DESCRIBING THE STORE BEING
C          SEPARATED AND THE FLIGHT CONDITIONS AND CALCULATES THE
C          TRAJECTORY
C
C     THE PARENT AIRCRAFT MODEL PRODUCED BY PROGRAM 1 IS READ
C          FROM TAPE 12
C
C     DIMENSION STATEMENTS
C
      DIMENSION HEAD(20),FVN(6,7),HEADE1(20),HEADE2(20)
      LOGICAL CIRSTR,ELLSTR,MLTFIN,LFIN
C
C     COMMON STATEMENTS -----------------------------------------------
C
      COMMON /BFORC / FBOD(5,5),ABOD(5),IBOD(5),LFIN(5),NBOD
      COMMON /BFORCE/ CNSB,CYSB,CLMSB,CLNSB,CLLSB
      COMMON /BOPTNS/ OPTNS(50)
      COMMON /CFORCE/ CNX(80),CYX(80),XCP(81),DRDXCP(81),RCP(81)
     *    ,UT(81),VT(81),WT(81)
      COMMON /CONFIG/ NFU,NPY,NSTRS,NRACK
      COMMON /CONSTS/ PI,DTOR
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *    ,CDC,XMOM,ESTRMX,DX,DELX,VSTORE,ESTLGC
      COMMON /DIMENS/ IDIM(130)
      COMMON /EFORC1/ CYEM1,CNEM1,CLLEM1,CLMEM1,CLNEM1
      COMMON /EFORC2/ CYEM2,CNEM2,CLLEM2,CLMEM2,CLNEM2
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,PHIROL,MSF,IPLNR,CLALPH
      COMMON /FLOW/ ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NMSEGO,NROLL
      COMMON /JECTOR/ NJECTR,FXS,FYS,FZS,RMX,RMY,RMZ
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1   ,IDEJ,IDF,ID2
      COMMON /NEWFOR/ NTHETA,DTHETA,THETAD(37),STHETA(37),CTHETA(37),
     1REFL,SREF,NCONPT
      COMMON /NUFLOW/ FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4
      COMMON /OUTINI/ XNOSEI,YNOSEI,ZNOSEI,XCGI,YCGI,ZCGI,XBASEI,YBASEI,
     1ZBASEI
      COMMON /STGEOM/ SLTHC(7),SRMAX(7),SIBCR(7),CSIBCR(7),SSIBCR(7),
     1                YBSO(7),X SO(7),ZBSO(7),NUMSTR(7),NSHAPE(7),
     3                NSHPT,MSHAPE(7),SPHIRR(7)
      COMMON /STORSD/ NXBODY,XBD(151),RBD(151),NPOLY,XEND(7),COEF(7,7),
     1 XBC(150),RBC(150),DRDXBC(150),NSORCE,DRDXBD(151)
      COMMON /THRUST/ NTPOLY,TEND(5),TC(5,6),FTHRUS,NTHRUS
      COMMON /TOTFOR/ CSIDE,CNORM,CROLL,CPITCH,CYAW
      COMMON /VRTXV / VVRTX(250),WVRTX(250),NVRTPL,NVRTX,VRTMAX
      COMMON          A(62500)
C             ALSO CHANGE DEFINITION OF NADIM TO MATCH  A  DIMENSION
      EQUIVALENCE (OPTNS(27),IPRTS)
      EQUIVALENCE (IDIM(47),NTAP7), (IDIM(13),KRAD), (IDIM(39),NADIM)
C
C     FORMAT STATEMENTS -----------------------------------------------
C
  701 FORMAT(18I5)
  702 FORMAT(1H1,36X,51HSUPERSONIC SIX-DEGREE-OF-FREEDOM TRAJECTORY PROG
     1RAM//)
  703 FORMAT(20A4)
  704 FORMAT(10X,20A4)
  705 FORMAT(/1H )
  706 FORMAT(8F10.0)
  707 FORMAT(10X,26HAIRCRAFT FLIGHT CONDITIONS/15X,17HANGLE OF ATTACK =,
     1F6.2,8H DEGREES/15X,19HFLIGHT PATH ANGLE =,F6.2,8H DEGREES/15X,13H
     2MACH NUMBER =,F5.2/15X,26HFREE STREAM MASS DENSITY =,F11.7,21H SLU
     3GS PER CUBIC FOOT/15X,22HFREE STREAM VELOCITY =,F8.2,16H FEET PER
     4SECOND//)
  709 FORMAT(/20X,33HTHIS STORE HAS A PLANAR EMPENNAGE)
  710 FORMAT(/20X,36HTHIS STORE HAS A CRUCIFORM EMPENNAGE)
  711 FORMAT(20X,24HTHE EMPENNAGE FORCES ACT,F9.5,17H FEET BEHIND NOSE/
     120X,50HTHE AVERAGE BODY RADIUS IN THE EMPENNAGE REGION IS,F9.5,5H
     2FEET)
  712 FORMAT(5X,78HANGLE OF ATTACK AND/OR MACH NUMBER READ IN DOES NOT A
     1GREE WITH VALUE FROM FILE//28X,5HVALUE,5X,4HFILE/29X,4HREAD,4X,
     25HVALUE/10X,15HANGLE OF ATTACK,3X,F5.2,4X,F5.2/10X,11HMACH NUMBER,
     33X,2F5.2)
  714 FORMAT(20X,26HAXIAL-FORCE COEFFICIENT IS,F10.5)
  715 FORMAT(23X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTION/
     126X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X,
     22HC7)
  716 FORMAT(1H1,9X,12HSTORE NUMBER,I5,21H IS THE STORE EJECTED
     1    /15X,26HSTORE OPTIONS (0=NO,1=YES),
     2    ,/20X,40HCAPTIVE-STORE TRAJECTORY               NGAM =,I3
     3    ,/20X,40HROLLING MOMENT CALCULATED             NROLL =,I3
     4    ,/20X,40HNUMBER OF EMPENNAGES PRESENT           NEMP =,I3
     5    ,/20X,40HDAMPING INCLUDED IN FORCES            NDAMP =,I3
     6    ,/20X,40HTHRUST TIME HISTORY SPECIFIED        NTHRUS =,I3
     7    ,/20X,40HEJECTOR FORCE HISTORY SPECIFIED NJECTR =,I3)
  717 FORMAT(15X,31HADDITIONAL INPUT FOR THIS STORE/20X,12HSTORE MASS =,
     1F10.5,6H SLUGS/20X,45HMOMENTS AND PRODUCTS OF INERTIA, SLUG - SQ F
     2T/23X,5HIXX =,F10.5/23X,5HIYY =,F10.5/23X,5HIZZ =,F10.5/23X,5HIYZ
     3=,F10.5/23X,5HIXZ =,F10.5/23X,5HIXY =,F10.5)
  718 FORMAT(20X,22HSTORE MOMENT CENTER IS,F9.5,17H FEET BEHIND NOSE/20X
     1,55HSTORE CENTER OF GRAVITY OFFSET FROM MOMENT CENTER, FEET/23X,6H
     2XBAR =,F9.5/23X,6HYBAR =,F9.5/23X,6HZBAR =,F9.5)
  719 FORMAT(20X,34HPOLYNOMIALS SPECIFYING STORE SHAPE/
     123X,26HX/L OF END OF EACH SECTION/26X,7HSECTION,5X,3HX/L)
  720 FORMAT(28X,I2,3X,7F10.5)
  721 FORMAT(15X,50HCIRCULAR STORE BODY AND FORCE CALCULATION CONTROLS
     1    ,/20X,40HNUMBER OF BODY SEGMENTS                NSEG =,I3
     2    ,/20X,40HNUMBER OF SOURCES + DOUBLETS         NSORCE =,I3
     3    ,/20X,40HNUMBER OF BODY SHAPE POLYNOMIALS NPOLY =,I3
     4    ,/20X,40HNUMBER OF THETA STATIONS             NTHETA =,I3)
  729 FORMAT(20X,52HTHE TAIL FIN SEMISPAN MEASURED FROM THE BODY AXIS IS
     1,F9.5,5H FEET/20X,29HTHE FINS ARE INITIALLY ROLLED,F6.2,41H DEGREE
     2S FROM THE VERTICAL AND HORIZONTAL)
  735 FORMAT(20X,27HTHE FIN LIFT-CURVE SLOPE IS,F9.5,11H PER RADIAN)
  750 FORMAT(/20X,48HPOLYNOMIALS SPECIFYING STORE THRUST TIME HISTORY/
     123X,30HTIME AT END OF EACH POLYNOMIAL/26X,21HPOLYNOMIAL  TIME, SEC
     2)
  751 FORMAT(25X,I6,4X,F11.5)
  752 FORMAT(23X,27HCOEFFICIENTS OF POLYNOMIALS/26X,10HPOLYNOMIAL,7X,2HC
     11,9X,2HC2,9X,2HC3,9X,2HC4,9X,2HC5,9X,2HC6)
  753 FORMAT(25X,I6,6X,6F11.3)
 754    FORMAT(I5,5F10.2,5L2)
 755    FORMAT (///20X18HFINNED BODY LAYOUT
     *     /18X,5HSTORE,3X,13HFIN INDICATOR,3X,6HENDING
     *     /14X,1HI,2X,9HX-STATION,13H T-FIN/F-BODY,4X,4HRING)
 756    FORMAT (10X,I5,F10.3,L12,I10)
 757    FORMAT(//10X,44HLEADING EDGE OF 1ST WING/EMPENNAGE (XWLE1) =,F10.3
     *     /10X,44HLEADING EDGE OF 2ND WING/EMPENNAGE (XWLE2) =,F10.3)
C
C       CONSTANTS -----------------------------------------------------
C
        PI=3.141592653
        DTOR=0.0174532925
        ACCG=32.174
        NEJSHP = 0
        CLLSB = 0.0
        FXS = 0.0
        FYS = 0.0
        FZS = 0.0
        RMX = 0.0
        RMY = 0.0
        RMZ = 0.0
C
        WRITE (6,702)
C
C       READ AND WRITE HEADING INFORMATION FROM FILE PRODUCED BY PROGRAM 1
C
        REWIND 12
        READ (12) NCARDS
        DO 1 J=1,NCARDS
```

Figure C-1(a)

```
      READ (12) HEAD
C
C     READ AND WRITE ADDITIONAL HEADING INFORMATION
C
C                                          *********** INPUT CARD ***********
    1 WRITE (6,704) HEAD
      READ (5,701) NCARDS
      DO 2 J=1,NCARDS
      READ (5,702) HEAD
    2 WRITE (6,704) HEAD
      WRITE (6,705)
C
C     READ REMAINDER OF PROGRAM 1 FILE
C
      CALL ROFILE
C
C     READ FLIGHT CONDITIONS
C
C                                          *********** INPUT CARD ***********
      READ (5,706) ALFA,GAMF,CMACH,RHO,VINF
      WRITE(6,707) ALFA,GAMF,CMACH,RHO,VINF
C
C     CHECK TO SEE THAT ALFA AND CMACH AGREE WITH VALUES READ FROM FILE
C
      ALFAR=ALFA*DTOR
      IF (ALFAR.EQ.ALFACR.AND.CMACH.EQ.FMACH) GO TO 3
      ALFAC=ALFACR/DTOR
      WRITE (6,712) ALFA,ALFAC,CMACH,FMACH
      STOP 712
    3 CONTINUE
C
C     INPUT ADDITIONAL INFORMATION REQUIRED TO DESCRIBE EJECTED STORE
C
      IF (NSTRS.EQ.0) GO TO 1000
C                                          *********** INPUT CARD ***********
      READ (5,701) NEJECT,NSEG,NSORCE,NGAM,NPOLY,NROLL,NEMP,NDAMP,NTHRUS
     1,NTHETA,NJECTR
      WRITE (6,716) NEJECT,NGAM,NROLL,NEMP,NDAMP,NTHRUS,NJECTR
      IF (NSEG+NSORCE+NTHETA+NPOLY.GT.0) WRITE(6,721) NSEG,NSORCE
     *    ,NPOLY,NTHETA
C
C     READ ADDITIONAL EJECTOR FORCE INPUT DATA  AND INITIALIZE IT
C
      IF (NJECTR.EQ.1) CALL EJECTR(DC,VAR,DTOR,TIME,SLTHC(NEJSTR)
     *    ,XMOM,NPOLY,COEF,XEND,ELLSTR)
C                                          *********** INPUT CARD ***********
      READ (5,706) SMASS,FIXX,FIYY,FIZZ,FIYZ,FIXZ,FIXY
      READ (5,706) XMOM,XBAR,YBAR,ZBAR
      WRITE (6,717) SMASS,FIXX,FIYY,FIZZ,FIYZ,FIXZ,FIXY
      WRITE (6,718) XMOM,XBAR,YBAR,ZBAR
      XMOM=ABS(XMOM)
      MLTFIN = NEMP.EQ.2
C
C  IDENTIFY EJECTED STORE SHAPE AND NUMBER
C
      NEJSHP = 0
      NESHP = 0
      DO 54 I=1,NSHPT
      ISHP = 0
      IF (MSHAPE(I).GT.50) NESHP=NESHP+1
      DO 53 J=1,NSTRS
      IF (NSHAPE(J).NE.MSHAPE(I)) GO TO 53
      ISHP = ISHP+1
      IF (NEJECT.NE.NUMSTR(J)) GO TO 53
      NEJSTR = J
      NEJSHP = NESHP
      NEJGB = ISHP
   53 CONTINUE
   54 CONTINUE
      CIRSTR = NSHAPE(NEJSTR).LE.50
      ELLSTR = NSHAPE(NEJSTR).GT.50
      IF (ELLSTR) GO TO 44
C                                          *********** INPUT CARD ***********
      READ (5,706) (XEND(J),J=1,NPOLY)
      DO 50 J=1,NPOLY
   50 READ (5,706) (COEF(J,K),K=1,7)
      WRITE (6,719)
      DO 51 J=1,NPOLY
   51 WRITE (6,720) J,XEND(J)
      WRITE (6,715)
      DO 52 J=1,NPOLY
   52 WRITE (6,720) J,(COEF(J,K),K=1,7)
C
C     DETERMINE GEOMETRIC PARAMETERS DESCRIBING CIRCULAR STORE
C
      ESTLGC=SLTHC(NEJSTR)
      ESTRMX=SRMAX(NEJSTR)
      SREF=PI*ESTRMX**2
      REFL=2.0*ESTRMX
      FSORCE=NSORCE
      DELX=ESTLGC/FSORCE
C
C     CALCULATE X AND R COORDINATES OF STORE BODY DEFINITION POINTS
C
      NXBODY=NSORCE+1
      XBD(1)=0.0
      RBD(1)=0.0
      CALL SHAPE (XBD(1),NPOLY,XEND,COEF,RR,DRDXBD(1))
      DO 40 J=2,NXBODY
      XBD(J)=XBD(J-1)+DELX
      XXX=XBD(J)/ESTLGC
      CALL SHAPE (XXX,NPOLY,XEND,COEF,RR,DRDXBD(J))
   40 RBD(J)=RR*ESTLGC
C
C     CALCULATE X, R, AND DRDX OF STORE BODY CONTROL POINTS
C
      DO 41 J=1,NSORCE
      XBC(J)=0.5*(XBD(J)+XBD(J+1))
      XXX=XBC(J)/ESTLGC
      CALL SHAPE (XXX,NPOLY,XEND,COEF,RR,DRDXBC(J))
   41 RBC(J)=RR*ESTLGC
C
C     CALCULATE THETAS TO BE USED IN FORCE AND MOMENT CALCULATION
C
      FTHETA=NTHETA
      DTHETA=2.0*PI/(FTHETA-1.0)
      THETA=-DTHETA
      DO 42 J=1,NTHETA
      THETA=THETA+DTHETA
      THETAD(J)=THETA/DTOR
      STHETA(J)=SIN(THETA)
   42 CTHETA(J)=COS(THETA)
C
C     CALCULATE X AND R FOR FORCE AND MOMENT CALCULATION
C
      FSEG=NSEG
      DELX=ESTLGC/FSEG
      XX=-0.5*DELX
      DO 43 J=1,NSEG
      XX=XX+DELX
      XCP(J)=XX
      XXX=XX/ESTLGC
      CALL SHAPE (XXX,NPOLY,XEND,COEF,RR,DUM)
   43 RCP(J)=RR*ESTLGC
      IF (NFU.EQ.2 .OR. NESHPT.GT.0) GO TO 44
      GO TO 45
C
C  SET UP GEOMETRY FOR ELLIPTIC STORE FOR SOURCE PANEL METHOD ----------
C
   44 CALL STRDAT (NFU,SREF,REFL,ESTLGC,CIRSTR)
      NADIM = 62500
      FSTRMX = 0.5*REFL
C
C
   45 CONTINUE
C                                          *********** INPUT CARD ***********
      READ (5,706) CA
      WRITE(6,714) CA
```

Figure C-1(b)

```
      NASYM=0
      IF (ABS(XBAR).GT.1.0E-05) NASYM=1
      IF (ABS(YBAR).GT.1.0E-05) NASYM=1
      IF (ABS(ZBAR).GT.1.0E-05) NASYM=1
      SAPG=SIN(ALFACR+GAMF*DTOR)
      CAPG=COS(ALFACR+GAMF*DTOR)
C
C     INPUT EMPENNAGE DATA IF EMPENNAGE IS PRESENT
C
      CNEM1 =0.0
      CYEM1 =0.0
      CLLEM1=0.0
      CLMEM1=0.0
      CLNEM1=0.0
      CYEM2 =0.0
      CNEM2 =0.0
      CLLEM2=0.0
      CLMEM2=0.0
      CLNEM2=0.0
      IF (NEMP.EQ.0) GO TO 150
      IF (ELLSTR) GO TO 110
C             SLENDER BODY METHOD (USED WITH CIRCULAR BODY OPTION ONLY)
C                                         *********** INPUT CARD ***********
      READ (5,701) IPLNR,MSF
      READ (5,706) XTAIL,RADAV,FINSS,PHIROL,CLALPH
      IF (IPLNR.EQ.1) WRITE (6,709)
      IF (IPLNR.EQ.0) WRITE (6,710)
      WRITE (6,711) XTAIL,RADAV
      WRITE (6,729) FINSS,PHIROL
      WRITE (6,735) CLALPH
      CALL SEMPIN
      GO TO 150
C
C     GENERATE EMPENNAGE PANELING AND INFLUENCE COEFFICIENTS
C
110   REWIND 3
      REWIND 10
C
C  READ INPUT DEFINING LOCATIONS OF AXIAL FIN BODY STATIONS
C
C                                         *********** INPUT CARD ***********
      READ (5,754) NBOD, (XBOD(I),I=1,5),(LFIN(I),I=1,5)
      CALL IXBOD (XBOD,IBOD,NBOD,LFIN,XWLE1,XWLE2,RA1,RA2,RB1,RB2)
C
      WRITE(6,755)
      WRITE(6,756) (I,XBOD(I),LFIN(I),IBOD(I),I=1,NBOD)
      WRITE(6,757) XWLE1,XWLE2
      NVRTX = 0
      CALL CRFWBD (FMACH,SREF,REFL,XWLE1,RA1,RB1,KRAD,MLTFIN,MEADE1)
      REWIND 7
      IF (MLTFIN) CALL IOREAD(7,A,NTAP7)
      IF (MLTFIN) CALL CRFWBD (FMACH,SREF,REFL,XWLE2,RA2,RB2
     *                ,KRAD,MLTFIN,MEADE2)
C
C     INPUT THRUST TIME HISTORY IF POWERED STORE
C
  150 IF (NTHRUS.EQ.0) GO TO 57
C                                         *********** INPUT CARD ***********
      READ (5,701) NTPOLY
      READ (5,706) (TEND(J),J=1,NTPOLY)
      DO 131 J=1,NTPOLY
  131 READ (5,706) (TC(J,K),K=1,6)
      WRITE (6,750)
      DO 132 J=1,NTPOLY
  132 WRITE (6,751) J,TEND(J)
      WRITE (6,752)
      DO 133 J=1,NTPOLY
  133 WRITE (6,753) J,(TC(J,K),K=1,6)
C
C     INITIALIZE FOR TRAJECTORY CALCULATION
C
   57 VAR(11)=SIBCR(NEJSTR)
      VAR(10)=0.0
      VAR(12)=SPHIRR(NEJSTR)
C                                         *********** INPUT CARD ***********
      READ (5,706) VXZERO,VYZERO,VZZERO,
     1            VAR(4),VAR(5),VAR(6)
      VAR(1)=VZZERO*SSIBCR(NEJSTR)+VXZERO*CSIBCR(NEJSTR)
      VAR(2)=VYZERO
      VAR(3)=VZZERO*CSIBCR(NEJSTR)-VXZERO*SSIBCR(NEJSTR)
C                                         *********** INPUT CARD ***********
      VAR(7)=XBSO(NEJSTR)-XMOM*CSIBCR(NEJSTR)
      VAR(8)=YBSO(NEJSTR)
      VAR(9)=ZBSO(NEJSTR)+XMOM*SSIBCR(NEJSTR)
      XNOSEI=XBSO(NEJSTR)
      YNOSEI=YBSO(NEJSTR)
      ZNOSEI=ZBSO(NEJSTR)
      XCGI=VAR(7)
      YCGI=VAR(8)
      ZCGI=VAR(9)
      XBASEI=XNOSEI-SLTHC(NEJSTR)*CSIBCR(NEJSTR)
      YBASEI=YNOSEI
      ZBASEI=ZNOSEI+SLTHC(NEJSTR)*SSIBCR(NEJSTR)
C                                         *********** INPUT CARD ***********
      READ (5,706) DTIME,TIMEI,TIMEF
      NEQ=12
      DDTIME=DTIME
      TIME=TIMEI
      IF (TIME.LE.0.0) GO TO 60
C                                         *********** INPUT CARD ***********
      READ (5,706) VAR
      DO 61 J=10,12
   61 VAR(J)=DTOR*VAR(J)
   60 NDIFEQ=1
C
C     EJECTOR FORCE CALCULATIONS PRIOR TO SEPARATION
C
      IF (NJECTR.EQ.2) CALL EJECTR(DC,VAR,DTOR,TIME,SLTHC(NEJSTR)
     *    ,XMOM,NPOLY,COEF,XEND,ELLSTR)
      CALL ADAMS (DTIME,DDTIME, VAR,DVAR,NEQ,NDIFEQ,TIME)
      NOUT=1
C
C     CALCULATE AERODYNAMIC FORCES AND MOMENTS -------------------------
C
   62 CONTINUE
C
C  COMPUTE FREE STREAM COMPONENT OF VELOCITY VECTOR
C
      CALL VXYZ
C             CIRCULAR STORE FORCE CALCULATIONS
      IF (CIRSTR) CALL SFORCE
      IF (NEMP.EQ.1 .AND. CIRSTR) CALL SEMFOR
C
C             ELLIPTICAL FORCE CALCULATIONS USING PANELING METHODS
C
      IF (.NOT.ELLSTR) GO TO 65
      CALL SFORC2
      REWIND 7
      IF (MLTFIN) CALL IOWRIT (7,A,NTAP7)
      REWIND 3
      REWIND 10
      NVRTX = 0
      IF (MLTFIN) CALL EGMRST
      IF (NEMP.GE.1) CALL DEMON2(VAR,DC,XMOM,VSTORE,CYEM1,XWLE1,MEADE1)
      REWIND 7
      IF (MLTFIN) CALL IOREAD(7,A,NTAP7)
      IF (MLTFIN .AND. NVRTX.GT.0) CALL VPATH (0,NVRTX,VRTMAX)
      IF (MLTFIN) CALL EGMRST
      IF (MLTFIN) CALL DEMON2(VAR,DC,XMOM,VSTORE,CYEM2,XWLE2,MEADE2)
   65 CONTINUE
      IF (NGAM.EQ.1) CALL DIRCOS(VAR,DC)
C
C     CALCULATE ACCELERATIONS ------------------------------------------
C
C     CALCULATE COEFFICIENT MATRIX
C
      DO 70 J=1,6
      DO 70 K=1,6
```

Figure C-1(c)

```
   70 FVN(J,K)=0.0
      FVN(1,1)=1.0
      FVN(2,2)=1.0
      FVN(3,3)=1.0
      FVN(4,4)=FIXX
      FVN(5,5)=FIYY
      FVN(6,6)=FIZZ
      FVN(4,5)=-FIXY
      FVN(4,6)=-FIXZ
      FVN(5,6)=-FIYZ
      FVN(5,4)=-FIXY
      FVN(6,4)=-FIXZ
      FVN(6,5)=-FIYZ
      IF (NASYM.EQ.0) GO TO 80
      FVN(1,4)=YBAR*DC(1,3)-ZBAR*DC(1,2)
      FVN(1,5)=ZBAR*DC(1,1)-XBAR*DC(1,3)
      FVN(1,6)=XBAR*DC(1,2)-YBAR*DC(1,1)
      FVN(2,4)=YBAR*DC(2,3)-ZBAR*DC(2,2)
      FVN(2,5)=ZBAR*DC(2,1)-XBAR*DC(2,3)
      FVN(2,6)=XBAR*DC(2,2)-YBAR*DC(2,1)
      FVN(3,4)=YBAR*DC(3,3)-ZBAR*DC(3,2)
      FVN(3,5)=ZBAR*DC(3,1)-XBAR*DC(3,3)
      FVN(3,6)=XBAR*DC(3,2)-YBAR*DC(3,1)
      DO 71 J=4,6
      DO 71 K=1,3
   71 FVN(J,K)=SMASS*FVN(K,J)
C
C     CALCULATE RIGHT HAND SIDE
C
   80 GXX=-ACCG*(SAPG*DC(1,1)-CAPG*DC(3,1))
      GYY=-ACCG*(SAPG*DC(1,2)-CAPG*DC(3,2))
      GZZ=-ACCG*(SAPG*DC(1,3)-CAPG*DC(3,3))
      QSTORE=0.5*RHO*VSTORE**2
      QSREF=QSTORE*SREF
      QSREFL=QSREF*REFL
C              ADD BODY LOAD DUE TO BODY WITHOUT PRESENCE OF EMPENNAGE(S)
      CNORM=CNSB
      CSIDE=CYSB
      CROLL=CLLSB
      CPITCH=CLMSB
      CYAW=CLNSB
      IF (NEMP.EQ.0) GO TO 84
      IF (CIRSTR) GO TO 83
C              SUBTRACT BODY FORCE CONTRIBUTIONS OF FINNED PORTION,
C              BODY FORCE IS DELETED AND INCLUDED IN LATER FIN
C              CALCULATIONS
      CALL FXBOD (CYX,CNX,FBOD,IBOD,NBOD)
      IEMP = 1
      DO 82 I=1,NBOD
      IF (.NOT.LFIN(I)) GO TO 82
      IF (IEMP.EQ.2) GO TO 81
      CYEM1 =CYEM1 -FBOD(1,I)
      CNEM1 =CNEM1 -FBOD(2,I)
      CLLEM1=CLLEM1-FBOD(3,I)
      CLMEM1=CLMEM1-FBOD(4,I)
      CLNEM1=CLNEM1-FBOD(5,I)
      IEMP=2
      GO TO 82
   81 CYEM2 =CYEM2 -FBOD(1,I)
      CNEM2 =CNEM2 -FBOD(2,I)
      CLLEM2=CLLEM2-FBOD(3,I)
      CLMEM2=CLMEM2-FBOD(4,I)
      CLNEM2=CLNEM2-FBOD(5,I)
   82 CONTINUE
C              ADD FORCE CONTRIBUTION DUE FORWARD SET OF FINS
   83 CSIDE=CSIDE+CYEM1
      CNORM=CNORM+CNEM1
      CROLL=CROLL+CLLEM1
      CPITCH=CPITCH+CLMEM1
      CYAW=CYAW+CLNEM1
      IF (NEMP.LT.2) GO TO 84
C              ADD FORCE CONTRIBUTION DUE AFT SET OF FINS
      CSIDE=CSIDE+CYEM2
      CNORM=CNORM+CNEM2
      CROLL=CROLL+CLLEM2
      CPITCH=CPITCH+CLMEM2
      CYAW=CYAW+CLNEM2
   84 CONTINUE
C
C     ADD THRUST FORCE AND MOMENTS
C
      FTHRUS=0.0
      IF (NTHRUS.NE.0) CALL THRCAL(TIME)
C
C     EJECTOR FORCES AND MOMENTS
C
      IF (NJECTR.EQ.3) CALL EJECTR(DC,VAR,DTOR,TIME,SLTHC(NEJSTR)
     *   ,XMOM,NPOLY,COEF,XEND,ELLSTR)
C
C
      RONE=GXX-QSREF*CA/SMASS
      RONE=RONE+FTHRUS/SMASS
      RONE=RONE+FXS/SMASS
      RTWO=GYY+QSREF*CSIDE/SMASS
      RTWO = RTWO+FYS/SMASS
      RTHR=GZZ-QSREF*CNORM/SMASS
      RTHR = RTHR+FZS/SMASS
      FVN(4,7)=QSREFL*CROLL
      FVN(4,7) = FVN(4,7)+RMX
      FVN(5,7)=QSREFL*CPITCH
      FVN(5,7) = FVN(5,7)+RMY
      FVN(6,7)=QSREFL*CYAW
      FVN(6,7) = FVN(6,7)+RMZ
      IF (NASYM.EQ.0) GO TO 85
      RONE=RONE+XBAR*(VAR(5)**2+VAR(6)**2)-YBAR*VAR(4)*VAR(5)-ZBAR*VAR(4
     1)*VAR(6)
      RTWO=RTWO-XBAR*VAR(4)*VAR(5)+YBAR*(VAR(4)**2+VAR(6)**2)-ZBAR*VAR(5
     1)*VAR(6)
      RTHR=RTHR-XBAR*VAR(4)*VAR(6)-YBAR*VAR(5)*VAR(6)+ZBAR*(VAR(4)**2+VA
     1R(5)**2)
      FVN(4,7)=FVN(4,7)+SMASS*(GZZ*YBAR-GYY*ZBAR)
      FVN(5,7)=FVN(5,7)+SMASS*(GXX*ZBAR-GZZ*XBAR)
      FVN(6,7)=FVN(6,7)+SMASS*(GYY*XBAR-GXX*YBAR)
   85 FVN(1,7)=RONE*DC(1,1)+RTWO*DC(1,2)+RTHR*DC(1,3)
      FVN(2,7)=RONE*DC(2,1)+RTWO*DC(2,2)+RTHR*DC(2,3)
      FVN(3,7)=RONE*DC(3,1)+RTWO*DC(3,2)+RTHR*DC(3,3)
      FVN(4,7)=FVN(4,7)-VAR(6)*VAR(5)*(FIZZ-FIYY)+(VAR(5)**2-VAR(6)**2)*
     1FIYZ+VAR(4)*(VAR(5)*FIXZ-VAR(6)*FIXY)
      FVN(5,7)=FVN(5,7)-VAR(6)*VAR(4)*(FIXX-FIZZ)+(VAR(6)**2-VAR(4)**2)*
     1FIXZ+VAR(5)*(VAR(6)*FIXY-VAR(4)*FIYZ)
      FVN(6,7)=FVN(6,7)-VAR(4)*VAR(5)*(FIYY-FIXX)+(VAR(4)**2-VAR(5)**2)*
     1FIXY+VAR(6)*(VAR(4)*FIYZ-VAR(5)*FIXZ)
C
C     SOLVE FOR ACCELERATIONS
C
      CALL INVER2(FVN,1,6,6,7)
      DO 86 J=1,6
   86 DVAR(J)=FVN(J,7)
      DO 87 J=7,9
   87 DVAR(J)=VAR(J-6)
      SPHI=SIN(VAR(12))
      CPHI=COS(VAR(12))
      DVAR(10)=(VAR(5)*SPHI+VAR(6)*CPHI)/COS(VAR(11))
      DVAR(11)=VAR(5)*CPHI-VAR(6)*SPHI
      DVAR(12)=VAR(4)+SIN(VAR(11))*DVAR(10)
C
C     OUTPUT TRAJECTORY DATA IF AT END OF INTEGRATION STEP
C
      IF (NOUT.EQ.0) GO TO 91
      CALL SOUTPT
      NOUT=0
      IF (TIME+1.0E-05-TIMEF) 91,1000,1000
C
C     CALL INTEGRATION ROUTINE
C
   91 CALL ADAMS (DTIME, DDTIME,VAR,DVAR,NEQ,NDIFEQ,TIME)
      IF (NDIFEQ.EQ.1) GO TO 1000
      IF (NDIFEQ.GT.7) NOUT=1
```

Figure C-1(d)

```
      GO TO 62
 1000 CONTINUE
      STOP
      END


      SUBROUTINE ADAMS(H,DS,Y,DY,NEQ,NDIFEQ,S)
C     ADAMS INTEGRATION ROUTINE
C      FIXED INTEGRATION INTERVAL
C
      DIMENSION Y(12),DY(1c)
      DIMENSION Y1(12),DY3(12),Y5(12),E(12),Y3(12),DY1(12),Y2(12),PX(12)
     1,PF(12),DY2(12)
      GO TO (100,200,300,400,500,600,700,800,900),NDIFEQ
C
C     START BY RUNGE-KUTTA
C
  100 H=DS
 1101 JB=1
      DO 101 I=1,NEQ
  101 Y1(I)=Y(I)
  102 SX1=S
  103 NDIFEQ=2
      RETURN
  200 DO 201 I=1,NEQ
      DY3(I)=DY(I)
      Y5(I)=Y(I)
      TEMP=H*DY(I)
      Y(I)=0.5*TEMP+Y5(I)
  201 E(I)=TEMP
      S=0.5*H+S
      NDIFEQ=3
      RETURN
  300 DO 301 I=1,NEQ
      TEMP=H*DY(I)
      Y(I)=0.5*TEMP+Y5(I)
  301 E(I)=E(I)+2.0*TEMP
      NDIFEQ=4
      RETURN
  400 DO 401 I=1,NEQ
      TEMP=H*DY(I)
      Y(I)=Y5(I)+TEMP
  401 E(I)=E(I)+2.0*TEMP
      S=0.5*H+S
      NDIFEQ=5
      RETURN
  500 DO 501 I=1,NEQ
  501 Y(I)=(H*DY(I)+E(I))*0.16666667+Y5(I)
      GO TO (502,507,509,902),JB
  502 CONTINUE
  507 DO 508 I=1,NEQ
      DY1(I)=DY3(I)
  508 Y2(I)=Y(I)
      JB=3
      GO TO 103
  509 S=S-H
C
C     OUTPUT OF RUNGE-KUTTA
C
      IF (JB-4) 521,802,802
  521 NDIFEQ=8
      DO 522 I=1,NEQ
      PX(I)=Y(I)
      PF(I)=DY(I)
  522 Y(I)=Y2(I)
      RETURN
  800 DO 801 I=1,NEQ
      Y(I)=PX(I)
      DY(I)=PF(I)
  801 DY2(I)=DY3(I)
      JB=4
      S=S+H
  802 NDIFEQ=9
      RETURN
  900 IF (JB-5) 103,901,9901
  901 CON=0.041666667*H
      JB=6
 9901 NDIFEQ=6
      GO TO 600
  902 JB=5
      GO TO 802
C
C     ADAMS INTEGRATION
C
  600 DO 601 I=1,NEQ
      Y1(I)=Y(I)
      Y(I)=Y(I)+CON*(55.0*DY(I)-59.0*DY3(I)+37.0*DY2(I)-9.0*DY1(I))
      DY1(I)=DY2(I)
      DY2(I)=DY3(I)
      DY3(I)=DY(I)
  601 Y3(I)=Y(I)
      S=S+H
      NDIFEQ=7
      RETURN
  700 DO 701 I=1,NEQ
  701 Y(I)=Y1(I)+CON*(9.0*DY(I)+19.0*DY3(I)-5.0*DY2(I)+DY1(I))
      GO TO 802
      END


      SUBROUTINE BDCOEF (XFP,YFP,ZFP,NFLD,BETA,XWLE,RA,RB)
C
C  ROUTINE TO ORGANIZE GENERATION OF INFLUENCE COEFFICIENT MATRICES ON
C  TAPE10 FOR STORE BODY AT FIELD POINTS XFP+XWLE,YFP,ZFP.
C
C  TO PREVENT NUMERICAL INACCURACIES ASSOCIATED WITH FIELD POINTS
C  LYING INSIDE BODY SOURCE PANELS, ALL BIP CONTROL POINTS ARE
C  EXTRAPOLATED TO THE ACTUAL ELLIPTIC SURFACE.  EXTRAPOLATION IS
C  PERFORMED BY PROJECTING CONTROL POINTS ALONG THE RADIAL LINE FROM
C  THE ORIGIN.
C
      DIMENSION XFP(NFLD),YFP(NFLD),ZFP(NFLD)
      LOGICAL SUBSON,SUPERS
      COMMON /BODCOM/ AMACH,TANO,CX,XCOR(4),YCOR(4),ZCOR(4)
     1   ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,J
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *    ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *    ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IAO,IDO,ISKO,IZZ(4),NRING,IROW(50)
      COMMON           A(12000)
C
      BETAO = BETA
      BETAL = BETA
      SUBSON = BETA.LT.0.0
C
C  USE SCRATCH AREA AT END OF BLANK COMMON  FOR TEMPORARY COEFFICIENT
C  MATRIX STORAGE.
C
      IXF = NTAP7+1
      IYF = IXF+NFLD
      IZF = IYF+NFLD
      ISVN = IZF+NFLD
      LSKP = ISVN+NFLD
      IUB = LSKP+NFLD
      NAFLD = IUB+3*NFLD*MAXKR
      IF (NAFLD.GT.NADIM) WRITE(6,712) NTAP7,IUB,NAFLD
712   FORMAT (/5X,40HWARNING - NAFLD EXCEEDS BLANK COMMON
     *    65X,12H** BDCOEF **,/6X,6HNTAP7=,I6,6H  IUB=,I6,8H  NAFLD=,I6)
C
C  GENERATE TEMPORARY ARRAY OF X-COORDINATES
C
      JXF = IXF-1
      DO 100 I=1,NFLD
```

Figure C-1(e)

```
      JXF = JXF+1
      JYF = JXF+NFLD
      JZF = JYF+NFLD
      A(JXF) = XFP(I)+XWLE
      A(JYF) = YFP(I)
      A(JZF) = ZFP(I)
      CALL ELLCPT (A(JYF),A(JZF),RA,RB)
  100 CONTINUE
C
      CALL FLDAIC (A(I?? ,A(IYF),A(IZF),A(ISVN),A(LSKP),A(IUB),NFLD)
      RETURN
      END


      SUBROUTINE BDUVW (BDU,BDV,BDW,NFLD)
C
C  ROUTINE TO ORGANIZE CALCULATION OF U,V,W FROM COEFFICIENT MATRICES
C  STORED ON TAPE10
C
      DIMENSION BDU(NFLD),BDV(NFLD),BDW(NFLD)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *   ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IAO,IDO,ISKO,IZ2(4),NRING,IROW(50)
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1  ,IDEJ,IDF,ID2
      COMMON          A(12000)
C
      JGB = IGB(NEJGB)
      IUB = NTAP7+1
      CALL FLDUVW (A(IUB),A(JGB),BDU,BDV,BDW,NFLD,IROW,NRING)
      RETURN
      END


      SUBROUTINE BDYPR(NDAMP,XM,VSTORE,VAR)
C
C     VERSION&DEMON2.
C
C     THIS SUBROUTINE COMPUTES LINEAR AND BERNOULLI PRESSURES AT POINTS
C     ON THE INTERFERENCE SHELL.
C     WHERE THE BODY
C     IS COVERED WITH INTERFERENCE PANELS, THE POINTS COINCIDE WITH
C     THE PANEL CONTROL POINTS.  THE BODY MERIDIANS ON WHICH THE
C     POINTS LIE PASS THROUGH THE CONTROL POINTS OF THE BODY
C     INTERFERENCE PANELS.  THE NUMBER OF BODY MERIDIANS IS EQUAL TO
C     NBOCR, THAT IS, THE NUMBER OF BODY INTERFERENCE PANELS ON THE
C     BODY CIRCUMFERENCE.
C
C
      DIMENSION VAR(12)
C
C
C
      LOGICAL BODY
C
C
      COMMON /BVEL  / BDU(250),BDV(250),BDW(250)
      COMMON/ICVEL/UTCHK,VTCHK,WTCHK,IIT,IFT,MJ
      COMMON /ONE   / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRB(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLR(250),ZRR(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFRIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,IC,MSWR,MSWL,
     7   MSLU,MSWO,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
      COMMON /SPCPRS/ DLTP(250)
      COMMON/SPSANG/SINALC,COSALC,SINPHI,COSPHI
      COMMON/SWEEPS/VSWLER(20),VSWTER(20),VSWLEL(20),VSWTEL(20),
     1   VSWLEU(20),VSWTEU(20),VSWLED(20),VSWTED(20),LVSWP,LEFT,FAC,K?W8,
     2AVPNL(250),WIDTH(250)
      COMMON /VRTXV / VVRTX(250),WVRTX(250),NVRTPL,NVRTX,VRTMAX
      COMMON/WBTR/THTI(100),XWLE
C
C
C
      FACTR1=1.428571429/(FMACH*FMACH)
      FACTR2=0.2*FMACH*FMACH
      NOCPT=1
C
C
C     CALCULATE PRESSURE COEFFICIENTS AT CONTROL POINTS OF BODY
C     INTERFERENCE PANELS.
C     CONTROL POINT COORDINATES ARE IN WING SYSTEM.
C
C
C
C     CALCULATE CONTRIBUTION FROM CONSTANT U-VELOCITY PANELS ON BODY
C     INTERFERENCE SHELL AND WINGS OR FINS.
C     ADD IN VELOCITIES INDUCED BY MOVING VORTICES(IF APPLICABLE).
C
C
      IBD=0
      II=1
      IF=NWBP
      NRING=NBIP/NCWB
      FACTR1=1.428571429/(FMACH*FMACH)
      FACTR2=0.2*FMACH*FMACH
      DO 15 IL=1,NCWB
      DO 17 JL=1,NRING
      IBD=IBD+1
      IC=NPANLS+IBD
      XBDY=XCPT(IC)+XWLE
      CALL VELNOR (XCPT(IC),YCPT(IC),ZCPT(IC))
C
C     ADD EFFECTS OF 2-D TYPE VORTICES IF APPLICABLE
C     THEY HAVE BEEN CALCULATED ALREADY IN CRFWBD.
C     (BY MEANS OF SUBROUTINE VRTVEL OR VVELS)
C
      UTOT=UCHK+BDU(IC)
      VTOT=VCHK+BDV(IC)+VVRTX(IC)
      WTOT=WCHK+BDW(IC)+WVRTX(IC)
C
C     ADD IN CONTRIBUTION DUE TO PITCHING, YAWING AND ROLLING MOTION
C
      IF (NDAMP.NE.0) CALL VELDMP(UTOT,VTOT,WTOT,XBDY,YCPT(IC),ZCPT(IC)
     *   ,1,XM,VSTORE,VAR)
      PRESSL=-2.0*UTOT
      BDUSQ=UTOT*UTOT
      BDVSQ=VTOT*VTOT
      BDWSQ=WTOT*WTOT
      UBAR=UTOT*COSALC-VTOT*SINBET*WTOT*SINALF
      ARG=1.0-FACTR2*(2.0*UBAR+BDUSQ+BDVSQ+BDWSQ)
      PRESSB=-FACTR1
      IF(ARG.GE.TOTLR) PRESSB=FACTR1*(ARG**3.5-1.0)
      DLTP(IC) = PRESSB
   17 CONTINUE
   15 CONTINUE
      RETURN
      END


      SUBROUTINE BVARIA(XB,YB,ZB)
C
C     THIS SUBROUTINE CALCULATES THE BETAS TO BE USED FOR
C          AXISYMMETRIC AND NONCIRCULAR FUSELAGE, RACK, AND ALL
C          CIRCULAR AND ELLIPTIC STORE BODIES, EXCEPT THE SEPARATED
C          STORE
C
      DIMENSION SX(50),SR(50),IA(1)
```

Figure C-1(f)

```
      COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,DBETA
     *   ,EALPHA,CNUO,CNU2,XSHLDR,SHK(3),XSHK(100),RSHK(100)
      COMMON /CONFIG/ NFU,NPY,NSTRS,NRACK
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *   ,CDC,XMON,ESTRMX,DX,DELX,VSTORE,ESTLGC
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITM,IDEL,NTAPT
     *   ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IAO,IDO,ISKO,IZZ(4),NWING,IROW(50)
      COMMON /FLOW/ ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101),
     1     FRMAX,FLTHC,NFPOLY,FXEND(7),FCOEF(7,7)
      COMMON /FSOR/ FXL(101),FSS(100),FDS(100),NFSOR
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NMSEGO,NROLL
      COMMON /LOCBET/ FBETAL,RBETAL,SBETAL(7)
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1  ,IDEJ,IDF,ID2
      COMMON /RKGEOM/ RLTHC,RRMAX,RIBCR,CRIBCR,SRIBCR,YBRO,XBRO,ZBRO
      COMMON /RSHOCK/ NRSHK,RXSHK(50),RRSHK(50),RDRDX(101)
      COMMON /RSOR/ NRSOR,RXL(101),RSS(100),RDS(100)
      COMMON /SSHOCK/ NSSHK(7),SXSHK(50,7),SRSHK(50,7),SDRDX(101,7)
      COMMON /SSOR/ SXL(101,7),SSS(100,7),SDS(100,7),NSSOR(7)
      COMMON /STGEOM/ SLTHC(7),SRMAX(7),SIBCR(7),CSIBCR(7),SSIBCR(7),
     1                YBSO(7),XBSO(7),ZBSO(7),NUMSTR(7),NSHAPE(7)
     3                ,NSHPT,MSHAPE(7),SPHIRR(7)
      COMMON /STORSD/ NXBODY,XBD(151),RBD(151),NPOLY,XEND(7),COEF(7,7),
     1 XBC(150),RBC(150),DRDXBC(150),NSORCE,DRDXBD(151)
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
      COMMON          A(12000)
      EQUIVALENCE (IA(1),A(1))
C
C     FUSELAGE ************* CIRCULAR BODY OPTION **********************
C
      IF (NFU.EQ.0) GO TO 50
      RADIUS=SQRT(YB*YB+ZB*ZB)
      IF (NFU.NE.1) GO TO 20
C
C     ADJUST SHOCK FOR ANGLE OF ATTACK
C
      CPHI=ZB/RADIUS
      ALFA=CPHI*ALFACR
      SALF=SIN(ALFA)
      CALF=COS(ALFA)
      DO 2 J=1,NFSHK
      X=FXSHK(J)
      R=FRSHK(J)
      SX(J)=X*CALF+R*SALF
    2 SR(J)=R*CALF-X*SALF
C
C     LOCATE SHOCK WAVE
C
      CALL SHKLOC(XS,RADIUS,NFSHK,SX,SR)
      GO TO 30
C
C  NONCIRCULAR FUSELAGE SHOCK SHAPE --------------------------------------
C  SHOCK WILL BE ASSUMED TO CONTAIN ANGLE OF ATTACK VARIATION
C  FROM PROGRAM 1.
C        DEFINE SHOCK SHAPE VARIABLE LOCATIONS IN BLANK COMMON
C
 20   IF (NFU.NE.2) GO TO 50
      ISK = IA(IDF+75)
      INSHK = ISK+1
      IPHIS = ISK+11
      IXSHK = ISK+41
      IRSHK = ISK+141
      NSHOCK = IA(ISK+32)
      FXSHLD =  A(ISK+37)
C
      CALL XVSRT (XS,ABS(YB),-ZB,IA(INSHK),A(IPHIS)
     *   ,NSHOCK,A(IXSHK),A(IRSHK),0.0,0.0,.FALSE.)
C
C     CALCULATE BETAL
C
   30 FBETAL=BETA
      BTNOSE=XS/RADIUS
   32 XINFTY=FXSHLD+BETA*RADIUS
      FMSHK=SQRT(BTNOSE*BTNOSE+1.0)
      XX=-XB
      IF (XX.LT.XS) GO TO 36
      IF (XX.GT.XS) GO TO 34
      FBETAL=BTNOSE
      GO TO 36
   34 IF (XX.LT.XINFTY) GO TO 35
      GO TO 36
   35 FML=FMSHK+(FMACH-FMSHK)*(XX-XS)/(XINFTY-XS)
      FBETAL=SQRT(FML*FML-1.0)
   36 CONTINUE
C
C     RACK ***************************************************************
C
   50 IF (NRACK.EQ.0) GO TO 100
C
C     LOCATE POINT IN RACK SYSTEM
C
      TA=XB-XBRO
      TB=ZB-ZBRO
      XR=TA*CRIBCR-TB*SRIBCR
      YR=YB-YBRO
      ZR=TA*SRIBCR+TB*CRIBCR
      RADIUS=SQRT(YR*YR+ZR*ZR)
C
C     ADJUST SHOCK FOR ANGLE OF ATTACK
C
      CPHI=ZR/RADIUS
      ALFA=CPHI*(ALFACR+RIBCR)
      SALF=SIN(ALFA)
      CALF=COS(ALFA)
      DO 52 J=1,NRSHK
      X=RXSHK(J)
      R=RRSHK(J)
      SX(J)=X*CALF+R*SALF
   52 SR(J)=R*CALF-X*SALF
C
C     LOCATE SHOCK WAVE
C
      CALL SHKLOC(XS,RADIUS,NRSHK,SX,SR)
C
C     CALCULATE BETAL
C
      RBETAL=BETA
      BTNOSE=XS/RADIUS
      XINFTY =RXSHLD+BETA*RADIUS
      FMSHK=SQRT(BTNOSE*BTNOSE+1.0)
      XX=-XR
      IF (XX.LT.XS) GO TO 56
      IF (XX.GT.XS) GO TO 54
      RBETAL=BTNOSE
      GO TO 56
   54 IF (XX.LT.XINFTY) GO TO 55
      GO TO 56
   55 FML=FMSHK+(FMACH-FMSHK)*(XX-XS)/(XINFTY-XS)
      RBETAL=SQRT(FML*FML-1.0)
   56 CONTINUE
C
C     STORES *************************************************************
C
  100 IF (NSTRS.LE.1) RETURN
C
C     STORE LOOP
C
      DO 200 K=1,NSTRS
      IF (K.EQ.NEJSTR) GO TO 200
C
C     LOCATE POINT IN STORE K SYSTEM
C
      DX=XB-XBSO(K)
      DZ=ZB-ZBSO(K)
```

Figure C-1(g)

```
      CSI=CSIBCR(K)
      SSI=SSIBCR(K)
      XST=DX*CSI-DZ*SSI
      YST=YB-YBSO(K)
      ZST=DX*SSI+DZ*CSI
      RADIUS=SQRT(YST*YST+ZST*ZST)
      IF (NSHAPE(K).GT.50) GO TO 120
C
C     ADJUST SHOCK FOR ANGLE OF ATTACK
C
      CPHI=ZST/RADIUS
      ALFA=CPHI*(ALFACR+SIBCR(K))
      SALF=SIN(ALFA)
      CALF=COS(ALFA)
      JMAX=NSSHK(K)
      DO 102 J=1,JMAX
      X=SXSHK(J,K)
      R=SRSHK(J,K)
      SX(J)=X*CALF+R*SALF
  102 SR(J)=R*CALF-X*SALF
C
C     LOCATE SHOCK WAVE
C
      CALL SHKLOC(XS,RADIUS,JMAX,SX,SR)
      GO TO 130
C
C  ELLIPTIC STORE SHOCK SHAPE
C  DEFINE LOCATION OF SHOCK SHAPE VARIABLES IN BLANK COMMON OR /BSHOCK/
C
C  DEFINE Y AND Z FOR ROLLED STORE BODY
120   CONTINUE
      PHI =SPHIRR(K)
      CPHI=COS(PHI)
      SPHI=SIN(PHI)
      DY=YST
      DZ=ZST
      YST= DY*CPHI+DZ*SPHI
      ZST=-DY*SPHI+DZ*CPHI
      IF (NSHAPE(K).NE.NSHAPE(NEJSTR)) GO TO 125
C        USE EJECTED STORE SHAPE DATA
      ALP = (ALFACR+SIBCR(K))*(1.0-EALPHA)
      CALL XVSRT (XS,YST,-ZST,NSHK,PHIS,NSHOCK,XSHK,RSHK,ALP,PHI,.TRUE.)
      GO TO 130
C        USE 2ND STORE SHAPE DATA
125   CONTINUE
      ISK = IA(ID2+75)
      INSHK = ISK+1
      IPHIS = ISK+11
      IXSHK = ISK+41
      IRSHK = ISK+141
      NSHOCK = IA(ISK+32)
      SXSHLD(K) = A(ISK+37)
      ALP = (ALFACR+SIBCR(K))*(1.0-A(ISK+34))
      CALL XVSRT (XS,YST,-ZST,IA(INSHK),A(IPHIS)
     *   ,NSHOCK,A(IXSHK),A(IRSHK),ALP,PHI,.TRUE.)
C
C     CALCULATE BETAL
C
130   SBETAL(K)=BETA
      BTNOSE=XS/RADIUS
      XINFTY=SXSMLD(K)+BETA*RADIUS
      FMSHK=SQRT(BTNOSE*BTNOSE+1.0)
      XX=-XST
      IF (XX.LT.XS) GO TO 106
      IF (XX.GT.XS) GO TO 104
      SBETAL(K)=BTNOSE
      GO TO 106
  104 IF (XX.LT.XINFTY) GO TO 105
      GO TO 106
  105 FML=FMSHK+(FMACH-FMSHK)*(XX-XS)/(XINFTY-XS)
      SBETAL(K)=SQRT(FML*FML-1.0)
  106 CONTINUE
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CEL1(RES,AK,IER)
C
C     ..................................................................
C
C        SUBROUTINE CEL1
C
C        PURPOSE
C           CALCULATE COMPLETE ELLIPTIC INTEGRAL OF FIRST KIND
C
C        USAGE
C           CALL CEL1(RES,AK,IER)
C
C        DESCRIPTION OF PARAMETERS
C           RES   - RESULT VALUE
C           AK    - MODULUS (INPUT)
C           IER   - RESULTANT ERROR CODE WHERE
C                   IER=0  NO ERROR
C                   IER=1  AK NOT IN RANGE -1 TO +1
C
C        REMARKS
C           FOR AK=+1,-1 THE RESULT IS SET TO 1.E30.
C           FOR MODULUS AK AND COMPLEMENTARY MODULUS CK,
C           EQUATION AK*AK+CK*CK=1.0 IS USED.
C           AK MUST BE IN THE RANGE -1 TO +1
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           CEL1(AK)=INTEGRAL(1/SQRT((1+T*T)*(1+(CK*T)**2))), SUMMED
C           OVER T FROM 0 TO INFINITY).
C           EQUIVALENT ARE THE DEFINITIONS
C           CEL1(AK)=INTEGRAL(1/(COS(T)SQRT(1+(CK*TAN(T))**2))+SUMMED
C           OVER T FROM 0 TO PI/2).
C           CEL1(AK)=INTEGRAL(1/SQRT(1-(AK*SIN(T))**2),SUMMED OVER T
C           FROM 0 TO PI/2), WHERE K=SQRT(1.-CK*CK).
C           EVALUATION
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R.BULIRSCH, *NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS
C           AND ELLIPTIC FUNCTIONS*, HANDBOOK SERIES SPECIAL FUNCTIONS,
C           NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ..................................................................
C
      IER=0
C
C        TEST MODULUS
C
      GEO=1.-AK*AK
      IF(GEO)1,2,3
    1 IER=1
      RETURN
C
C        SET RESULT VALUE =OFLOW
C
    2 RES=1.E30
      RETURN
    3 GEO=SQRT(GEO)
      ARI=1.
    4 AARI=ARI
      TEST=AARI*1.E-4
      ARI=GEO+ARI
C
C        TEST OF ACCURACY
C
      IF(AARI-GEO-TEST)6,6,5
```

Figure C-1(h)

```
5 GEO=SQRT(AARI*GEO)
  ARI=0.5*ARI
  GO TO 4
6 RES=3.14159265/ARI
  RETURN
  END



  SUBROUTINE CEL2(RES,AK,A,B,IER)
C
C ..................................................................
C
C    SUBROUTINE CEL2
C
C    PURPOSE
C       COMPUTES THE GENERALIZED COMPLETE ELLIPTIC INTEGRAL OF
C       SECOND KIND.
C
C    USAGE
C       CALL CEL2(RES,AK,A,B,IER)
C
C    DESCRIPTION OF PARAMETERS
C       RES   - RESULT VALUE
C       AK    - MODULUS (INPUT)
C       A     - CONSTANT TERM IN NUMERATOR
C       B     - FACTOR OF QUADRATIC TERM IN NUMERATOR
C       IER   - RESULTANT ERROR CODE WHERE
C               IER=0  NO ERROR
C               IER=1  AK NOT IN RANGE -1 TO +1
C
C    REMARKS
C       FOR AK = +1,-1 THE RESULT VALUE IS SET TO 1.E30 IF B IS
C       POSITIVE, TO -1.E30 IF B IS NEGATIVE.
C       SPECIAL CASES ARE
C       K(K) OBTAINED WITH A = 1, B = 1
C       E(K) OBTAINED WITH A = 1, B = CK*CK WHERE CK IS
C       COMPLEMENTARY MODULUS.
C       B(K) OBTAINED WITH A = 1, B = 0
C       D(K) OBTAINED WITH A = 0, B = 1
C       WHERE K, E, B, D DEFINE SPECIAL CASES OF THE GENERALIZED
C       COMPLETE ELLIPTIC INTEGRAL OF SECOND KIND IN THE USUAL
C       NOTATION, AND THE ARGUMENT K OF THESE FUNCTIONS MEANS
C       THE MODULUS.
C
C    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C       NONE
C
C    METHOD
C       DEFINITION
C       RES=INTEGRAL((A+B*T*T)/(SQRT((1+T*T)*(1+(CK*T)**2))*(1+T*T))
C       SUMMED OVER T FROM 0 TO INFINITY).
C       EVALUATION
C       LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C       REFERENCE
C       R.BULIRSCH, *NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS
C       AND ELLIPTIC FUNCTIONS*, HANDBOOK SERIES SPECIAL FUNCTIONS,
C       NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C ..........................    ....................................
C
C
  IER=0
C
C    TEST MODULUS
C
  GEO=1.-AK*AK
  IF(GEO)1,2,6
1 IER=1
  RETURN
C
C    SET RESULT VALUE = OVERFLOW
C
2 IF(B)3,5,4
3 RES=-1.E30
  RETURN
4 RES=1.E30
  RETURN
5 RES=A
  RETURN
C
C    COMPUTE INTEGRAL
C
6 GEO=SQRT(GEO)
  ARI=1.
  AA=A
  AN=A+B
  W=B
7 W=W+AA*GEO
  W=W+W
  AA=AN
  AARI=ARI
  ARI=GEO+ARI
  AN=W/ARI+AN
C
C    TEST OF ACCURACY
C
  IF(AARI-GEO-1.E-4*AARI)9,9,8
8 GEO=SQRT(GEO*AARI)
  GEO=GEO+GEO
  GO TO 7
9 RES=.78539816*AN/ARI
  RETURN
  END




  SUBROUTINE CRFWBD (AMACH,REFA,REFD,XLE,RAZ,RBY,KRAD,MLTFIN,HEAD)
C
C    VERSIONBDEMON2.
C
C    MULTI-FIN WING-BODY PROGRAM, SUPERSONIC FLOW
C    SET UP GEOMETRY, COORDINATES, AERODYNAMIC INFLUENCE COEFFICIENT
C    MATRIX FOR EMPENNAGES WITH FROM 1 TO 4 FINS
C
  DIMENSION YRT(20),YLT(20),HEAD(20),ZUT(20),ZDT(20)
C
  LOGICAL BODY,DELTA,MLTFIN
  LOGICAL FIN1,FIN2,FIN3,FIN4,NOFIN1,NOFIN2,NOFIN3,NOFIN4
C
  COMMON /BVEL  / BDU(250),BDV(250),BDW(250)
  COMMON /CONSTS/ PI,DTOR
  COMMON /INTPDT/ PHIDIH,THETIT,PHIFR,PHIFL,PHIFU,PHIFD
  COMMON/OAFM/XM,ZM,CZOA,CYOA,CMOA,CLNOA,CLLOA
  COMMON /ONE   / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
 1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
 2   ,XLF(250),XLR(250),XRF(250),XRB(250),YLC(250),YRC(250),ZLF(250)
 3   ,ZRF(250),ZLB(250),ZRB(250),SNT(100),CST(100),SNT2(100),
 4   CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
 5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
 6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
 7   MSWU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
 8   RA,RB,ERATIO,BODY,DELTA
  COMMON/SWEEPS/VSWLER(20),VSWTER(20),VSWLEL(20),VSWTEL(20),
 1  VSWLEU(20),VSWTEU(20),VSWLED(20),VSWTED(20),LVSWP,LEFT,FAC,NCWB,
 2ARPNL(250),WIDTH(250)
  COMMON/THREE/ANGLR,ANGLL,ANGLU,ANGLD,DELR,DELL,DELU,DELD,SREF,REFL
  COMMON /VRTXV / VVRTX(250),WVRTX(250),NVRTIN,NVRTX,VRTMAX
  COMMON/WBTR/THT1(100),XWLE
  COMMON          FVN(12000)
C
C
    1 FORMAT (20A4)
  700 FORMAT (1H1,15(1H*),20A4,15(1H*),10H CRFWBD **)
  703 FORMAT( // 18X,52H              PANEL CORNER COORDINATES FOR WING PA
     1NELS)
```

Figure C-1(i)

```
  706 FORMAT(3X,I3,3X,F9.4,1X,F9.4,1X,F9.4,1H*,F9.4,1X,F9.4,1X,F9.4,
     1  1H*,F9.4,1X,F9.4,1X,F9.4,1H*,F9.4,1X,F9.4,1X,F9.4)
  708 FORMAT(/1X,58(1H-),13HWING 1 PANELS     ,59(1H-)/)
  709 FORMAT(/1X,58(1H-),13HWING 2 PANELS     ,59(1H-)/)
  710 FORMAT(/1X,58(1H-),13HWING 3 PANELS     ,59(1H-)/)
  711 FORMAT(/1X,58(1H-),13HWING 4 PANELS     ,59(1H-)/)
  713 FORMAT(3F10.5)
  726 FORMAT(///25X,36HBIP CORNER COORDINATES IN WING FRAME)
 7030 FORMAT (                                  //5X,1HJ,15X,2HLF,15X,
     1  1H*,12X,2HLB,15X,1H*,12X,2HRF,15X,1H*,12X,2HRB/12X,1HX,9X,1HY,
     2  9X,1HZ,5X,1H*,3X,1HX,9X,1HY,9X,1HZ,5X,1H*,3X, 1HX,9X,1HY,9X,1HZ,
     3  5X,1H*,3X,1HX,9X,1HY,9X,1HZ/38X,1H*,29X,1H*,29X,1H*/130(1H*)/
     4  38X,1H*,29X,1H*,29X,1H*)
  732 FORMAT (3X,69HSPECIFIED SPANWISE LOCATIONS OF OUTBOARD PANEL EDGES
     1 AND SWEEP ANGLES//2X,1HK,4X,6HY OR Z,7X,10HL.E. SWEEP,5X,
     2 10HT.E. SWEEP/23X,5HANGLE,10X,5HANGLE///)
  733 FORMAT (1X,I2,2X,F10.5,2(5X,F10.5))
  734 FORMAT (1H1,4X,14HWING 1 SURFACE //)
  735 FORMAT (1H1,4X,14HWING 2 SURFACE //)
  736 FORMAT (1H1,4X,14HWING 3 SURFACE //)
  737 FORMAT (1H1,4X,14HWING 4 SURFACE //)
  740 FORMAT(//15X,21HINTERDIGITATED ANGLES
     *   /7X,20HPHIDIM          THETIT  ,/5X,2G12.5)
  741 FORMAT (//15X,15HBODY PROPERTIES //
     *    40H  VERTICAL SEMI PRINCIPAL AXIS   (RA) = ,F10.4/
     *    40H  HORIZ. SEMI PRINCIPAL AXIS     (RB) = ,F10.4/
     *    40H  ELLIPTIC RATIO             (ERATIO) = ,F10.4/
     *    40H  INTERFERENCE BODY LENGTH      (BIL) = ,F10.4/
     *    40H  NO. OF PANELS AROUND BODY   (NBDCR) = ,I5/
     *    40H  NO. OF STREAMWISE BODY PANELS(NCWB) = ,I5)
  742 FORMAT( /15X,9HREFERENCE,/ 7X,16HSREF          REFL,/5X,2G12.5)
  744 FORMAT (/15X,8HOPTIONAL
     *   /7X,26HNVRTIN    LVSWP     VRTMAX,/1X,2I10,G12.5)
  750 FORMAT(20I5)
  751 FORMAT(8F10.2)
  752 FORMAT(/20X,36HEMPENNAGE INPUT GEOMETRY DESCRIPTION,
     *   /15X,12HFIN PROPERTY,15X,10HFIN 1 OR R,5X,10HFIN 2 OR L
     *   ,5X,10HFIN 3 OR U,5X,10HFIN 4 OR D//)
  753 FORMAT (40H  NO. OF PANELS - SPANWISE       (MSW) = ,I7,3I15,/
     *    40H  ROOT CHORD                      (CR) = ,F10.3,3F15.3,/
     *    40H  LEADING EDGE SWEEP            (SWLE) = ,F10.3,3F15.3,/
     *    40H  TRAILING EDGE SWEEP           (SWTE) = ,F10.3,3F15.3,/
     *    40H  EXPOSED SEMISPAN                (B2) = ,F10.3,3F15.3,/
     *    40H  FIN DIMEDRAL                  (PHIF) = ,F10.3,3F15.3,/
     *    40H  BODY ANGLE OF FIN ATTACHMENT  (THET) = ,F10.3,3F15.3,/
     *    40H  FIN DEFLECTION                 (DEL) = ,F10.3,3F15.3)
C
C
      SREF = REFA
      REFL = REFD
      XWLE = XLE
      RA = RAZ
      RB = RBY
      ERATIO = RB/RA
C
C
C     DEFAULT VALUES FOR NAMELIST INPUT
C
      BIL=0.0
      B2V=0.0
      CRPV=0.0
      FAC=0.95
      LVSWP=0
      NOUT=0
      NPR=0
      NVRTIN=0
      SWLEP=0.0
      SWLEV=0.0
      SWTEP=0.0
      SWTEV=0.0
      XM=0.0
      ZM=0.0
C
C*********************************************************************
C
C     WING GEOMETRY SPECIFIED IN PLANFORM,
C     GEOMETRICALLY SYMMETRIC WINGS ONLY
C
C
C     READ HEADER CARD
C
      READ(5,1) HEAD
C
C     READ INPUT
C
      READ (5,750) MSWR,MSWL,MSWU,MSWD,NCW,NCWB,NBDCR,LVSWP
      IF (MSWR+MSWL.GT.0) READ(5,751) CRP,SWLEP,SWTEP,B2
      IF (MSWU+MSWD.GT.0) READ(5,751) CRPV,SWLEV,SWTEV,B2V
      READ(5,751) PHIDIM,THETIT,DELR,DELL,DELU,DELD
      IF (THETIT.LE.0.0 .OR. THETIT.GE.90.)
     *   READ(5,751) PHIFR,PHIFL,PHIFU,PHIFD,THETR,THETL,THETU,THETD
      READ(5,751) BIL,VRTMAX
      READ(5,750) NVRTIN,NOUT,NPR
      IF (NBDCR.LE.0) NBDCR=KRAD-1
      IF (VRTMAX.LE.0.) VRTMAX=0.35
C
C
C         CHECK FOR INTERDIGITATED TAIL DIMEDRAL ANGLES
C
C  NOTE* IF INTERDIGITATED FINS ARE UNDER CONSIDERATION,THE FOLLOWING
C     CORRELATION HOLDS.
C
C         RIGHT UPPER FIN.....RIGHT OR 1
C         LEFT  LOWER FIN.....LEFT OR 2
C         RIGHT LOWER FIN.....UPPER OR 3
C         LEFT  UPPER FIN.....LOWER OR DOWN OR 4
C     THIS DESIGNATION ESTABLISHES CORRESPONDENCE BETWEEN CRUCIFORM AND
C     INTERDIGITATED FINS.
C  NOTE*THE DIHEDRAL ANGLE DEFINED AS THE CANT ANGLE HERE.
C
      IF (THETIT.LE.0.0 .OR. THETIT.GE.90.) GO TO 22
C
C         PHIFR = UPPER RIGHT CANT ANGLE.   PHIFL = LOWER LEFT CANT ANGLE
C         PHIFU = LOWER RIGHT CANT ANGLE.   PHIFD = UPPER LEFT CANT ANGLE
C
      PHIFR= PHIDIM
      PHIFL= PHIDIM
      PHIFU=-PHIDIM
      PHIFD=-PHIDIM
C
C
C     THETR,THETU DEFINE LOCATIONS OF RHS FINS ON BODY CIRCUMFERENCE
C                 MEASURED POS. FROM +YW-AXIS COUNTERCLOCKWISE.
C     THETL,THETD DEFINE LOCATIONS OF LHS FINS ON BODY CIRCUMFERENCE
C                 MEASURED POS. FROM -YW-AXIS COUNTERCLOCKWISE.
C
      THETR= THETIT
      THETL= THETIT
      THETU=-THETIT
      THETD=-THETIT
   22 CONTINUE
C
C
C     LOGICAL VARIABLE TRUSYM MAKES USE OF GEOMETRIC AND LOADING
C     SYMMETRY PROPERTIES
C
C     BODY=NBDCR.NE.0
      DELTA=DELR.NE.0..OR.DELL.NE.0..OR.DELU.NE.0..OR.DELD.NE.0.
C
      NRP=NCW*MSWR
      NRP1=NRP+1
      NHP=NCW*(MSWR+MSWL)
      NHP1=NHP+1
      N3P=NCW*(MSWL+MSWR+MSWU)
      N3P1=N3P+1
      NPANLS=NCW*(MSWR+MSWL+MSWU+MSWD)
      NPANLP=NPANLS+1
      NBIP=NCWB*NBDCR
```

Figure C-1(j)

```
      NWBP=NPANLS-NBIP
      IF (NWBP.GT.8) NPR=8
C
      WRITE (6,700) HEAD
C
      FMACH = AMACH
      BETA=SQRT(ABS(FMACH*FMACH-1.) )
      TLRNC=(B2*15.E-5)**2
      TOTLR=2.0*TLRNC
      CONST= 4.0*PI
C
C
      MSWRP=MSWR+1
      MSWLP=MSWL+1
      MSWUP=MSWU+1
      MSWDP=MSWD+1
      FIN1=MSWR.GT.0
      FIN2=MSWL.GT.0
      FIN3=MSWU.GT.0
      FIN4=MSWD.GT.0
      NOFIN1=.NOT.FIN1
      NOFIN2=.NOT.FIN2
      NOFIN3=.NOT.FIN3
      NOFIN4=.NOT.FIN4
      IF (LVSWP.NE.0) GO TO 32
C
C     SET UP GEOMETRICALLY SYMMETRIC LAY-OUT FOR NO BREAKS IN SWEEP
C
      SWPLER=SWLEP
      SWPTER=SWTEP
      SWPLEL=SWLEP
      SWPTEL=SWTEP
      CRPT=CRP
      SPANR=B2
      SPANL=B2
      SWPLEU=SWLEV
      SWPTEU=SWTEV
      SWPLED=SWLEV
      SWPTED=SWTEV
      CRPTV=CRPV
      SPANU=B2V
      SPAND=B2V
      WRITE(6,752)
      WRITE(6,753) MSWR,MSWL,MSWU,MSWD,
     *   CRP,CRP,CRPV,CRPV,
     *   SWPLER,SWPLEL,SWPLEU,SWPLED,
     *   SWPTER,SWPTEL,SWPTEU,SWPTED,
     *   SPANR,SPANL,SPANU,SPAND,
     *   PHIFR,PHIFL,PHIFU,PHIFD,
     *   THETR,THETL,THETU,THETD,
     *   DELR,DELL,DELU,DELD
C
C
      SLPWTR= TAN(SWPTER*DTOR)
      SLPWLR= TAN(SWPLER*DTOR)
      SLPWLL= TAN(SWPLEL*DTOR)
      SLPWTL= TAN(SWPTEL*DTOR)
      SLPWLU=TAN(SWPLEU*DTOR)
      SLPWLD=TAN(SWPLED*DTOR)
      SLPWTU=TAN(SWPTEU*DTOR)
      SLPWTD=TAN(SWPTED*DTOR)
      GO TO 40
C
C     READ IN NON UNIFORM DISTANCES TO PANEL OUTBOARD EDGES.
C     IF WITH A BODY, MEASURE FROM FIN ROOT CHORD.
C
   32 READ (5,713) (YRT(KJ),VSWLER(KJ),VSWTER(KJ),KJ=1,MSWRP)
      SWPLER=VSWLER(MSWRP)
      SWPTER=VSWTER(MSWRP)
      SWPLEU=SWPLER
      SWPTEU=SWPTER
      SPANR=YRT(MSWRP)-YRT(1)
      SPANL=SPANR
      SPANU=SPANR
      IF (NOFIN2) GO TO 34
      READ (5,713) (YLT(KJ),VSWLEL(KJ),VSWTEL(KJ),KJ=1,MSWLP)
      SWPLEL=-VSWLEL(MSWLP)
      SWPTEL=-VSWTEL(MSWLP)
      SPANL=-(YLT(MSWLP)-YLT(1))
C
C     IF TRUSYM IS TRUE AND INTERDIGITATED TAIL IS TREATED,
C     DATA IS EXPECTED FOR THE RIGHT UPPER AND LOWER FINS
C
   34 IF (NOFIN3) GO TO 36
      READ(5,713) (ZUT(KJ),VSWLEU(KJ),VSWTEU(KJ),KJ=1,MSWUP)
      SWPLEU=VSWLEU(MSWUP)
      SWPTEU=VSWTEU(MSWUP
      SPANU=ZUT(MSWUP)-ZUT(1)
   36 IF (NOFIN4) GO TO 132
      READ(5,713)  (ZOT(KJ),VSWLED(KJ),VSWTED(KJ),KJ=1,MSWDP)
      SWPLED=-VSWLED(MSWDP)
      SWPTED=-VSWTED(MSWDP)
      SPAND=-(ZOT(MSWDP)-ZOT(1))
      GO TO 132
   40 CONTINUE
C
C     EQUAL SPANWISE WIDTHS, CONSTANT SWEEPS
C     PANELS IN HORIZONTAL WING SURFACES
C
      YRT(1)=0.0
      YRT(MSWRP)=SPANR+YRT(1)
      DYR=SPANR/FLOAT(MSWR)
      DO 100 I=2,MSWR
      AI=I-1
  100 YRT(I)=DYR*AI+YRT(1)
C
      IF (NOFIN2) GO TO 112
      YLT(1)=0.0
      YLT(MSWLP)=-SPANL+YLT(1)
      DYL=SPANL/FLOAT(MSWL)
      DO 110 I=2,MSWL
      AI=I-1
  110 YLT(I)=-DYL*AI+YLT(1)
C
C     PANELS IN VERTICAL PLANE
C
  112 IF (NOFIN3) GO TO 122
      ZUT(1)=0.0
      ZUT(MSWUP)=SPANU+ZUT(1)
      DZU=SPANU/FLOAT(MSWU)
      DO 120 I=2,MSWU
      AI=I-1
  120 ZUT(I)=DZU*AI+ZUT(1)
C
  122 IF (NOFIN4) GO TO 132
      ZOT(1)=0.0
      ZOT(MSWDP)=-SPAND+ZOT(1)
      DZD=SPAND/FLOAT(MSWD)
      DO 130 I=2,MSWD
      AI=I-1
  130 ZOT(I)=-DZD*AI+ZOT(1)
  132 CONTINUE
C
      WRITE(6,741) RA,RR,ERATIO,BIL,NHDCR,NCWR
      WRITE(6,740) PHIDIH,THETIT
      WRITE(6,742) SREF,REFL
      WRITE(6,744) NVRTIN,LVSWP,VRTMAX
      IF (LVSWP.EQ.0) GO TO 79
      WRITE (6,734)
      WRITE (6,732)
      WRITE (6,733) (K,YRT(K),VSWLER(K),VSWTER(K),K=1,MSWRP)
      IF (FIN2) WRITE (6,735)
      IF (FIN2) WRITE (6,733) (K,YLT(K),VSWLEL(K),VSWTEL(K),K=1,MSWLP)
      IF (FIN3) WRITE (6,736)
      IF (FIN3) WRITE (6,733) (K,ZUT(K),VSWLEU(K),VSWTEU(K),K=1,MSWUP)
      IF (FIN4) WRITE (6,737)
      IF (FIN4) WRITE (6,733) (K,ZOT(K),VSWLED(K),VSWTED(K),K=1,MSWDP)
   79 CONTINUE
```

Figure C-1(k)

```
C
C
C     LAY-OUT ELEMENTAL PANELS ON THE FINS.
C
C    NOTE* DISTANCES YRT,YLT,ZUT,ZDT ARE MEASURED FROM THE FIN ROOT
C     CHORDS IF BODY IS PRESENT
C
C
      CALL LAYOUT(SLPWLR,SLPWTR,YRT,MSWRP,CRPT,1,CTP,PHIFR,THETR)
      IF (FIN3.OR.FIN4) CTPV=CTP
C
      IF (FIN2)
     *   CALL LAYOUT(SLPWLL,SLPWTL,YLT,MSWLP,CRPT,2,CTPL,PHIFL,THETL)
C
      IF (FIN3)
     *   CALL LAYOUT(SLPWLU,SLPWTU,ZUT,MSWUP,CRPTV,3,CTPV,PHIFU,THETU)
C
      IF (FIN4)
     *   CALL LAYOUT(SLPWLD,SLPWTD,ZDT,MSWDP,CRPTV,4,CTPVD,PHIFD,THETD)
C
C     LAY-OUT BODY INTERFERENCE PANELS
C
      IF(NBDCR.EQ.0) GO TO 53
      CALL LAYBIP (NBDCR,BIL)
   53 CONTINUE
C
C
      WRITE(6,700) HEAD
      WRITE(6,703)
      WRITE(6,7030)
      WRITE(6,708)
      WRITE(6,706) (J,XLF(J),YLC(J),ZLF(J),XLB(J),YLC(J),ZLB(J),XRF(J),
     1  YRC(J),ZRF(J),XRB(J),YRC(J),ZRB(J),J=1,NRP)
C
      IF (NOFIN2) GO TO 71
      WRITE(6,709)
      WRITE(6,706) (J,XLF(J),YLC(J),ZLF(J),XLB(J),YLC(J),ZLB(J),XRF(J),
     1  YRC(J),ZRF(J),XRB(J),YRC(J),ZRB(J),J=NRP1,NHP)
C
   71 IF (NOFIN3) GO TO 72
      WRITE(6,710)
      WRITE(6,706) (J,XLF(J),YLC(J),ZLF(J),XLB(J),YLC(J),ZLB(J),XRF(J),
     1  YRC(J),ZRF(J),XRB(J),YRC(J),ZRB(J),J=NHP1,N3P)
C
   72 IF (NOFIN4) GO TO 73
      WRITE(6,711)
      WRITE(6,706) (J,XLF(J),YLC(J),ZLF(J),XLB(J),YLC(J),ZLB(J),XRF(J),
     1  YRC(J),ZRF(J),XRB(J),YRC(J),ZRB(J),J=N3P1,NPANLS)
C
   73 IF(NBDCR.EQ.0) GO TO 78
      WRITE(6,726)
      WRITE(6,7030)
      WRITE (6,706) (J,XLF(J),YLC(J),ZLF(J),XLB(J),YLC(J),ZLB(J),XRF(J),
     1YRC(J),ZRF(J),XRB(J),YRC(J),ZRB(J),J=NPANLP,NWBP)
   78 CONTINUE
C
C
C     GENERATE SOURCE PANEL INFLUENCE COEFFICIENT MATRIX DUE TO STORE
C     BODY
C
      CALL BDCOEF (XCPT,YCPT,ZCPT,NWBP,BETA,XWLE,RA,RB)
C
C
C
C     SET UP INFLUENCE COEFFICIENT MATRIX
C     IT IS THE LHS OF THE FLOW TANGENCY CONDITION.
C     NOTE* THE B.C. STATES THAT VELOCITY NORMAL TO THE PANELS MUST
C     BE ZERO.
C     THEREFORE ALL VELOCITIES MUST BE TRANSFORMED INTO INDIVIDUAL
C     FLAP COORDINATE SYSTEM.
C
C
C     CALCULATE THE INFLUENCE OF EACH PANEL ON EACH CONTROL POINT
C     II AND IF ARE THE LIMITS FOR SUMMATION OF THE INFLUENCE FUNCTION
C     IN SUBROUTINE VELNOR
C     IF II = IF, THE INFLUENCE OF A SINGLE PANEL AT A POINT RESULTS
C     AS IN THE COMPUTATION OF THE ARRAY FVN.  IF II=1 AND IF=NWBP,
C     THE INFLUENCES ARE SUMMED OVER ALL PANELS, AS IN THE COMPUTATION
C     OF RESULTANT VELOCITIES BELOW,...
C
C     --- I IS THE INDEX OF THE INFLUENCING PANEL
C
C
      JJ=0
      NOCPT=0
      DO 450 I=1,NWBP
      II=I
      IF=I
C
C     J IS THE INDEX OF THE INFLUENCED PANEL,I.E. ITS CONTROL POINT.
C
      DO 425 J=1,NWBP
      JB=J-NPANLS
      CALL VELNOR(XCPT(J),YCPT(J),ZCPT(J))
      JJ=JJ+1
      IF(J.LE.NPANLS) GO TO 417
C
C     INFLUENCED PANEL IS ON THE BODY INTERFERENCE SHELL.
C
      CALL ROTWB (VCHK,WCHK,VV,WW,JB)
      GO TO 418
C
C     INFLUENCED PANEL IS ON A FIN.
C
  417 IF (J.LE.NRP) PHIF=PHIFR*DTOR
      IF (FIN2.AND.(J.GT.NRP.AND.J.LE.NHP)) PHIF=PHIFL*DTOR
      IF (FIN3.AND.(J.GT.NHP.AND.J.LE.N3P)) PHIF=PHIFU*DTOR
      IF (FIN4.AND.(J.GT.N3P.AND.J.LE.NPANLS)) PHIF=PHIFD*DTOR
      CALL ROTWF(VCHK,WCHK,VV,WW,PHIF)
  418 FVN(JJ)=WW
  425 CONTINUE
  450 CONTINUE
      CALL PAS001 (FVN,NWBP,NER,NWBP)
      IF (NER.GT.1) STOP 001
      NAGAIN=NAGAIN+1
C             SAVE FIN GEOMETRIC ARRAYS FOR MULTIPLE FIN CASES
      IF (MLTFIN) CALL EGMSAV
C             SAVE INFLUENCE COEFFICIENT MATRIX
      JMAX = JJ
      CALL IOWRIT (3,FVN,JMAX)
      RETURN
      END
```

```
      SUBROUTINE DASCRU (A,B,H,N,X0,WK,IER,ES)
C
C     THIS SUBROUTINE PERFORMS INTEGRATION
C
C                                  ES SHOULD BE SET TO .5 TIMES THE
C                                  DESIRED RELATIVE PRECISION OF
C                                  THE SOLUTION
C
      DIMENSION          WK(1),X0(1)
C
      INTEGER            SW
C
      LOGICAL            BE,BH,BR,BX
C
      DATA               ZERO,PS,OPS,THREE,FOUR/0.,.5,1.5,3.,4./
C
      IER = 0
      IF(A - B) 4,100,4
    4 IB1=N+N
      IB2=IB1+N
      HMIN=0.01*ABS(H)
      BH=.TRUE.
```

Figure C-1(1)

```
      BR=.TRUE.
      BX=.TRUE.
C
C                                   CHECK FOR THE PROPER SIGN OF H
C
      H=SIGN(ABS(H),B-A)
      X=A
    5 XS=X
      DO 10  J=1,N
         IJK0=N+J
         WK(IJK0)=X0(J)
   10 CONTINUE
   15 HS=H
      Q=X+H-B
      BE=.TRUE.
      IF(.NOT.((H.GT.ZERO.AND.Q.GE.ZERO).OR.(H.LT.ZERO.AND.Q.LE.ZERO)))
     1     GO TO 20
      H=B-X
      BR=.FALSE.
   20 H3=H/THREE
C
C                                   CALCULATE SOLN. AT X+H
C   NOTES ARRAY WK CONTAINS Y FOR ODD INDEX,W FOR EVEN INDEX.
C
C
      DO 90  SW=1,5
         CALL F(X0,X,N,WK)
         DO 70  I=1,N
            Q=H3*WK(I)
            IJK0=N+I
            IJK1=IB1+I
            IJK2=IB2+I
            GO TO (25,30,35,40,45),SW
   25       R=Q
            WK(IJK1)=Q
            GO TO 50
   30       R=P5*(Q+WK(IJK1))
            GO TO 50
   35       R=THREE*Q
            WK(IJK2)=R
            R=.375*(R+WK(IJK1))
            GO TO 50
   40       R=WK(IJK1)+FOUR*Q
            WK(IJK1)=R
            R=OP5*(R-WK(IJK2))
            GO TO 50
   45       R=P5*(Q+WK(IJK1))
            Q= ABS(R+R - OP5*(Q+WK(IJK2)))
   50       X0(I)=WK(IJK0)+R
            IF(SW.NE.5) GO TO 70
C
C                                   AUTOMATIC STEP CHANGE
C
            E=ABS(X0(I))
            R=ES
            IF(E.GE.1.E-3) R=E*ES
C
C                                   TEST ADJUSTMENT OF THE STEP
C
            IF(Q.LT.R .OR.(.NOT. BX)) GO TO 65
            BR=.TRUE.
            BM=.FALSE.
            H=P5*H
            IF(ABS(H).GE.HMIN) GO TO 55
C
C                                   THE STEP IS HALVED RESTORE X AND X0.
C                                   AND GO BACK FOR REPEATED INTEGRATION
C                                   WITH THIS NEW STEP
C
            H=SIGN(1.,H)*HMIN
            BX=.FALSE.
   55       DO 60  J=1,N
               IJK0=N+J
               X0(J)=WK(IJK0)
   60       CONTINUE
            X=XS
            GO TO 15
   65       IF(Q.GE.0.03125*R) BE=.FALSE.
   70    CONTINUE
         GO TO (75,90,80,85,90),SW
   75    X=X+H3
         GO TO 90
   80    X=X+P5*H3
         GO TO 90
   85    X=X+P5*H
   90 CONTINUE
C
C                                   TEST A POSSIBLE DOUBLING OF THE STEP
C
      IF(.NOT.(BE.AND.BM.AND.BR)) GO TO 95
      H=H+H
      BX=.TRUE
   95 BM=.TRUE.
      IF(BR) GO TO 5
      H=HS
      IF(B.  OR. BE) GO TO 9005
      IER = 33
      GO TO 9005
  100 DO 105 I=1,N
         X0(I)=ZERO
  105 CONTINUE
 9005 RETURN
      END



      COMPLEX FUNCTION DBLU(Z)
C
C     VERSIONS DEMON2
C
C     THIS FUNCTION SUBROUTINE CALCULATES THE INTERMEDIATE TRANSFORM
C     VARIABLE W FOR THE CONFORMAL TRANSFORMATION OF AN ELLIPTICAL
C     BODY WITH WINGS
C     A DESCRIPTION OF THE METHODS AND EQUATIONS PROGRAMMED HERE
C     ARE GIVEN IN SECTION 5.1  OF APPENDIX I OF NASA CR-3122
C
      COMMON/COM1/A2,B2,R2
      COMMON/COM3/ZR,ZI
      COMMON/COM5/DWDZ
      COMMON/COM6/W2,W
C
      COMPLEX Z,Z2,DWDZ,W,W2,WW
C
      Z2=Z*Z
      ZR=REAL(Z)
      ZI=AIMAG(Z)
      IF(ZR.NE.0.0) ZR=ZR/ABS(ZR)
      IF(ZI.NE.0.0) ZI=ZI/ABS(ZI)
      Z2=Z2-A2+B2
      Y=AIMAG(Z2)
      AY=1.0
      IF(Y.LT.0.0) AY=-1.0
      AYZ=1.0
      IF(ZI.LT.0.0) AYZ=-1.0
      Z2=CSQRT(Z2)*AY*AYZ
      IF((ABS(ZI).LE.0.0).AND.(REAL(Z).LT.0.0)) Z2=CMPLX(-REAL(Z2),
     1      AIMAG(Z2))
      DWDZ=0.5*(1.0+Z/Z2)
      WW=0.5*(Z+Z2)
      W2=1.0/WW/WW
      DBLU=WW
      RETURN
      END
```

Figure C-1(m)

```
      SUBROUTINE DEMON2 (VAR,DC,XMOM,VSTORE,FCOEF,XLE,HEAD)

C   ROUTINE TO FORM THE RIGHT HAND SIDE OF WING,INTERFERENCE SHELL
C   VELOCITY EQUATIONS AND SOLVE FOR PRESSURES AND LOADS
C
      LOGICAL BODY,DELTA
      DIMENSION RHS(250),VAR(12),FCOEF(5),DC(3,3),HEAD(20)
      COMMON /BVEL  / BDU(250),BDV(250),BDW(250)
      COMMON /BVELFS/ VX,VY,VZ,VRATS,RVX,RVY,RVZ
      COMMON /CONSTS/ PI,DTOR
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NHSEGO,NROLL
      COMMON /INTHDT/ PHIDIH,THETIT,PHIFR,PHIFL,PHIFU,PHIFD
      COMMON /LETEM/ FMXT3,FMXT4
      COMMON /NUFLOW/FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4
      COMMON /OAFM  / XM,ZM,CZOA,CYOA,CMOA,CLNOA,CLLOA
      COMMON /ONE   / CELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRR(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLR(250),ZRR(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETAD,BETAR,CONST,
     5   CN,DX,EM,DMACH,SINALF,SINBET,SLOPE,TURNC,
     6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
     7   MSWU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,EPATIO,BODY,DELTA
      COMMON/SPSANG/SINALC,COSALC,SINPHI,COSPHI
      COMMON /TEVRT / GAMTE(20),           ,ZCG(20)
      COMMON/THREE/ANGLR,ANGLL,           SLD,DELR,DELL,DELU,DELD,SREF,REFL
      COMMON /VRTXV / VVRTX(25           (250),NVRTIN,NVRTX,VRTMAX
      COMMON /WBTR  / THTI(100),
      COMMON          FVN(12000)
C
C
      XM = XMOM
      ZM = 0.0
      XWLE = XLE
      XMXLE = XMOM-XWLE
C
C
C     INITIALIZE BODY INDUCED VELOCITIES AND VELOCITIES INDUCEL BY
C     VORTICES WHICH ARE FIXED IN THE CROSS FLOW PLANE.
C
      DO 10 KM=1,250
      VVRTX(KM)=0.0
      WVRTX(KM)=0.0
      BDU(KM)=0.0
      BDV(KM)=0.0
   10 BDW(KM)=0.0
C
C     COMPUTE BODY SOURCE PANEL INFLUENCE AT U-VELOCITY CONTROL POINTS
C     FROM COEFFICIENT MATRICES ON TAPE10.
C     BLANK COMMON IS USED FOR SOURCE PANEL DATA BEFORE BEING REPLACED
C     BY FVN ARRAY
C
      CALL BDUVW (BDU,BDV,BDW,NWBP)
C
C     DETERMINE EFFECTS OF CALCULATED TRAILING EDGE VORTICITY OF
C     FORWARD LIFTING SURFACES.  TREAT THEM AS FIXED VORTICES AT THE
C     CONSTANT U-VELOCITY PANEL CONTROL POINTS ON THE FINS AND BODY-
C     INTERFERENCE SHELL.
C     2-D VORTICES,CROSS FLOW PLANE VELOCITIES.
C
C
      IF (NVRTIN.GT.0) NVRTX=0
      IF (NVRTX.EQ.0) GO TO 40
      DO 35 IC= 1,NWBP
      YCP=YCPT(IC)
      ZCP=ZCPT(IC)
      CALL ELLCPT (YCP,ZCP,1.01*RA,1.01*RB)
   35 CALL VVELS (NVRTX,YCP,ZCP,YCG,ZCG,GAMTE,RA,RB,
     1VVRTX(IC),WVRTX(IC),VRTMAX)
   40 CONTINUE
C
C     SET UP SINGLE COLUMN MATRIX FOR RIGHT HAND SIDE
C     IT REPRESENTS THE EXTERNALLY INDUCED VELOCITIES AND ANGLE OF
C     PITCH AND SIDESLIP EFFECTS AS WELL AS FIN THICKNESS.
C   NOTE NO THICKNESS EFFECTS ACCOUNTED FOR IF INTERDIGITATED FINS
C     ARE CONSIDERED.
C   NOTES ANY VORTEX INDUCED VELOCITIES MUST BE REFERENCED W.R.T VSTORE
C
C     FREE STREAM ANGLE OF ATTACK PARAMETERS IN TERMS OF V AND W
C
      SINALF = RVZ
      SINBET = RVY
      SIN2 = SINALF*SINALF+SINBET*SINBET
      SINALC = SQRT(SIN2)
      COSALC = SQRT(1.-SIN2)
      ALFR  = ASIN(RVZ)
      ALFA  = ALFR/DTOR
      BETAR = ASIN( RVY)
      DMACH = FMACH/VRATS
C
C     ANGLES ANGLR,ANGLL,ANGLU,ANGLD GOOD ONLY FOR CRUCIFORM FINS OR
C     MONO-PLANE WING.
C
      RDELR=DELR*DTOR
      ANGLR=RDELR+ALFR
      RDELL=DELL*DTOR
      ANGLL=ALFR-RDELL
      RDELU=DELU*DTOR
      ANGLU=BETAR+RDELU
      RDELD=DELD*DTOR
      ANGLD=BETAR-RDELD
      SNDELR = SIN(RDELR)
      SNDELL = SIN(RDELL)
      SNDELU = SIN(RDELU)
      SNDELD = SIN(RDELD)
C
C     RIGHT HAND SIDE FOR WINGS OR FINS AT INCIDENCE TO BODY CENTERLINE
C
C
      DO 645 K=1,NWBP
      IF (K.GT.NPANLS) GO TO 603
      UEXT = 0.
      VEXT = 0.
      WEXT = 0.
      XCP = XM-XCPT(K)-XWLE
      CALL STTOIN (XCP,YCPT(K),-ZCPT(K),XI,ETA,ZETA,DC)
      XB = VAR(7)+XI
      YB = VAR(8)+ETA
      ZB = VAR(9)+ZETA
C
C   CALCULATE PERTURBATION VELOCITIES DUE TO PARENT A/C AND STORES
C
C
C     CALCULATE THE BETAS TO BE USED IN THE SOURCE-DOUBLET
C           VELOCITY CALCULATION FOR ALL AXISYMMETRIC BODIES EXCEPT THE
C           SEPARATED STORE
C
      CALL BVARIA(XB,YB,ZB)
C
C   CALCULATE PARENT AIRCRAFT LOCAL WING THICKNESS MACH NUMBER AT XB.
C
      IF (XB.GT.XT3 .OR. XB.LT.XT4) GO TO 140
      FNUMCH = FMXT3+(FMXT4-FMXT3)*(XB-XT3)/(XT4-XT3)
      BTSQNU = FNUMCH*FNUMCH-1.0
      BETANU = SQRT(BTSQNU)
      GO TO 145
  140 FNUMCH = FMACH
      BETANU = BETA
      BTSQNU = BETASQ
  145 CONTINUE
      CALL RESVEL (XB,YB,ZB,URES,VRES,WRES)
      CALL INTOST (URES,VRES,WRES,UEXT,VEXT,WEXT,DC)
C
C   ADD FREE STREAM COMPONENT AND CONVERT TO BODY AXES (F    -XB,YB,-ZB)
      UEXT = -VRATS*UEXT+RVX
      VEXT =  VRATS*VEXT+RVY
```

Figure C-1(n)

```
      WEXT = -VRATS*WEXT+RVZ
C
C  ADD STORE BODY SOURCE PANEL AND VORTEX VELOCITY COMPONENTS
      VEXT = VEXT+BOV(K)+VVRTX(K)
      WEXT = WEXT+BOW(K)+WVRTX(K)
C
C  ADD VELOCITIES DUE TO BODY SHOCK REFLECTION.  INFLUENCE IS
C  COMPUTED AS REAL STORE ON IMAGE STORE FIN CONTROL POINTS AND
C  TRANSFORMED.
C
      CALL IMAGFN (XCPT(K),YCPT(K),ZCPT(K),UIFIN,VIFIN,WIFIN,XWLE)
      UEXT = UEXT+UIFIN
      VEXT = VEXT+VIFIN
      WEXT = WEXT+WIFIN
C
C  ADD CONTRIBUTIONS DUE TO PITCHING,YAWING, AND ROLLING MOTION
C
      IF (NOAMP.NE.0) CALL
     *  VELOMP (UEXT,VEXT,WEXT,XCPT(K),YCPT(K),ZCPT(K),1
     *  ,XMXLE,VSTORE,VAR)
      IF(K.LE.NRP) GO TO 641
      IF(K.LE.NMP) GO TO 642
      IF(K.LE.N3P) GO TO 643
      IF(K.LE.NPANLS) GO TO 644
C           BODY INTERFERENCE PANELS
  603 RHS(K)=0.0
      GO TO 645
C           DEFINE RHS FOR RIGHT (HORIZONTAL) WING (FIN)
  641 PHIF=PHIFR*DTOR
      CALL ROTWF(VEXT,WEXT,VV,WW,PHIF)
      RHS(K)=-WW-SNDELP
      GO TO 645
C           DEFINE RHS FOR LEFT  (HORIZONTAL) WING (FIN)
  642 PHIF=PHIFL*DTOR
      CALL ROTWF(VEXT,WEXT,VV,WW,PHIF)
      RHS(K)=-WW-SNDELL
      GO TO 645
C           DEFINE RHS FOR UPPER (VERTICAL) FIN
  643 PHIF=PHIFU*DTOR
      CALL ROTWF(VEXT,WEXT,VV,WW,PHIF)
      RHS(K)=-WW-SNDELU
      GO TO 645
C           DEFINE RHS FOR LOWER (VERTICAL) FIN
  644 PHIF=PHIFO*DTOR
      CALL ROTWF(VEXT,WEXT,VV,WW,PHIF)
      RHS(K)=-WW-SNDELO
  645 CONTINUE
C
C*********************************************************************
      JMAX=NWBP**2
      CALL IOREAD (3,FVN,JMAX)
      CALL PAS002 (FVN,RHS,NWBP,1,NWBP,1)
      DO 630 I=1,NWBP
      DEMONP=RHS(I)*CONST
      DELTP(I)=DEMONP
      IF (ABS(DELTP(I)).LT.1.0E-10) DELTP(I)=0.0
  630 CONTINUE
C*********************************************************************
C
C
C     NON LINEAR PRESSURE LOADING CALCULATION ON FINS AND INTERFERENCE
C     SHELL.  NVRTPL=0 MEANS EXCLUDE VORTEX INDUCED VELOCITIES PARALLEL
C     TO SURFACES. THEN COMPUTE LOADS IN SUBR. LOADS CALLED BY SPECPR
C
C
      CALL SPECPR(NOAMP,XMXLE,VSTORE,VAR,HEAD)
C
C  COPY FORCE COEFFICIENTS INTO FORCE ARRAY
C
      FCOEF(1) = CYOA
      FCOEF(2) = CZOA
      FCOEF(3) = 0.0
      IF (NROLL.NE.0) FCOEF(3) = CLLOA
      FCOEF(4) = CMOA
      FCOEF(5) = CLNOA
      RETURN
      END


      SUBROUTINE DIRCOS (A,D)
C     SUBROUTINE TO CALCULATE DIRECTION COSINES
C
      DIMENSION A(12),D(3,3)
      SPSI=SIN(A(10))
      CPSI=COS(A(10))
      STHE=SIN(A(11))
      CTHE=COS(A(11))
      SPHI=SIN(A(12))
      CPHI=COS(A(12))
      D(1,1)=CTHE*CPSI
      D(1,2)=SPHI*STHE*CPSI-CPHI*SPSI
      D(1,3)=CPHI*STHE*CPSI+SPHI*SPSI
      D(2,1)=CTHE*SPSI
      D(2,2)=SPHI*STHE*SPSI+CPHI*CPSI
      D(2,3)=CPHI*STHE*SPSI-SPHI*CPSI
      D(3,1)=-STHE
      D(3,2)=SPHI*CTHE
      D(3,3)=CPHI*CTHE
      RETURN
      END


      COMPLEX FUNCTION DSDZ(S)
C
C     VERSION* DEMON2
C     A DESCRIPTION OF THE METHODS AND EQUATIONS PROGRAMMED HERE
C     ARE GIVEN IN SECTION 5.3  OF APPENDIX I OF NASA CR-3122
C     ROUTINE PERFORMS DERIVATIVE CALCULATION IN TRANSFORMATION.
C     SEE EQUATION I125, NASA CR-3122.
C
      COMMON/COM2/SIG2,H2
      COMMON/COM4/G2,G1
      COMMON/COM5/D,DZ
      COMMON/COM6/W2,W
      COMPLEX W,W2,D,DZ,G1,G2
      DSDZ=0.5*(1.0-SIG2*W2)*(1.0+G1/G2)*D*DZ
      RETURN
      END


      COMPLEX FUNCTION DZDS(S)
C
C     CALCULATE DZ/DZETA
C     SEE EQUATION I125, SECTION 5.3, APPENDIX I, NASA CR-3122.
C     VERSION* DEMON2
C
      COMMON/COM1/A2,B2,R2
      COMMON/COM2/SIG2,H2
      COMMON/COM6/W2,W
      COMPLEX W,S,W2,G1,S2,Z,Z2
      S2=S*S
      G1=0.5*(1.0-0.25*(A2-H2)*W2)*(1.0-R2/S2)
      Z=S+R2/S
      Z2=Z*Z-4.0*SIG2
      Y=AIMAG(Z)
      YZ=AIMAG(Z2)
      AY=1.0
      AYZ=1.0
      IF(Y.LT.0.0) AY=-1.0
      IF(YZ.LT.0.0) AYZ=-1.0
      Z2=CSQRT(Z2)*AY*AYZ
      IF((ABS(Y).LT.0.0).AND.(REAL(Z).LT.0.0)) Z2=CMPLX(-REAL(Z2),
```

Figure C-1(o)

```
      1       AIMAG(Z2))
      DZDS=G1*(1.0-Z/Z2)
      RETURN
      END


      COMPLEX FUNCTION D2SDZ2(S)
C
C     CALCULATE DZETA2/DZ2
C      SEE EQUATION I128, SECTION 5.3, APPENDIX I, NASA CR-3122.
C     VERSIONS DEMON2
C
      COMMON/COM2/SIG2,H2
      COMMON/COM4/G2,G1
      COMMON/COM5/DWDZ
      COMMON/COM6/W2,W
      COMPLEX W,W2,DWDZ,G2,G1
      D2SDZ2=(1.0-G1/G2)*SIG2*W2/W
      D2SDZ2=D2SDZ2-0.5*(1.0-SIG2*W2)*(1.0-SIG2*W2)*H2/(G2*(G1*G1-H2))
      D2SDZ2=D2SDZ2*DWDZ*DWDZ
      RETURN
      END


      SUBROUTINE EGMSAV
C
C ROUTINE TO SAVE DEMON2 EMPENNAGE GEOMETRIC ARRAY AND CONTROL
C VARIABLES ON TAPE3 FOR MULTIPLE SET OF FINS
C
      COMMON /INTRDT/ ANTRDT(6)
      COMMON /ONE   / AONE(5699)
      COMMON /THREE / ATHREE(10)
      COMMON /SWEEPS/ SWEEP(664)
      COMMON /WBTR  / AWBTR(101)
C
      CALL IOWRIT (3,AONE,5699)
      CALL IOWRIT (3,ATHREE,10)
      CALL IOWRIT (3,SWEEP,664)
      CALL IOWRIT (3,ANTRDT,6)
      CALL IOWRIT (3,AWBTR,101)
      RETURN
      END


      SUBROUTINE EGMRST
C
C ROUTINE TO RESTORE DEMON2 EMPENNAGE GEOMETRIC ARRAY AND CONTROL
C VARIABLES ON TAPE3 FOR MULTIPLE SET OF FINS
C
      COMMON /INTRDT/ ANTRDT(6)
      COMMON /ONE   / AONE(5699)
      COMMON /THREE / ATHREE(10)
      COMMON /SWEEPS/ SWEEP(664)
      COMMON /WBTR  / AWBTR(101)
C
      CALL IOREAD (3,AONE,5699)
      CALL IOREAD (3,ATHREE,10)
      CALL IOREAD (3,SWEEP,664)
      CALL IOREAD (3,ANTRDT,6)
      CALL IOREAD (3,AWBTR,101)
      RETURN
      END


      SUBROUTINE EJECTR(DC,VAR,DTOR,TIME,STLGTH,XMOM,
     1NPOLY,COEF,XEND,ELLSTR)
C
C     SUBROUTINE TO CALCULATE EJECTOR FORCES + MOMENTS ACTING ON
C     SEPARATING STORE.
C
C     ASSUMPTIONS$
C
C     1. EJECTOR MAY HAVE 1 OR 2 FEET.
C
C     2. THE EJECTOR STROKE IS IN A PLANE PERPENDICULAR TO THE
C     AIRCRAFT AXIS OF SYMMETRY ( THE INERTIAL X AXIS, SEE FIGURE
C     27 AFFDL-TR-74-130 VOLUME II ).
C
C     3. THE EJECTOR STOKE ACTS IN A PLANE CONTAINING THE STORE
C     LONGITUDINAL AXIS WHEN THE STORE IS IN THE CARRIAGE POSITION.
C
C     4. UP TO A 5TH ORDER POLYNOMIAL MAY BE USED TO REPRESENT THE
C     FORCE OF EACH EJECTOR FOOT. UP TO 5 POLYNOMIALS MAY BE USED
C     FOR EACH FOOT.
C
C     5. THE FOOT HAS NO VOLUME. I.E. THE EJECTOR IS ASSUMED TO BE
C     A LINE.
C
C     6. THE RACK/AIRCRAFT STRUCTURE IS RIGID,I.E. THE
C     RACK TRANSMITTS 100( OF THE FORCE TO THE STORE
C
C     OPTIONS
C
C     1. THE EJECTOR FORCE CAN BE SPECIFIED AS A FUNCTION OF EJECTOR
C     STROKE OR TIME.
C
C     NSTRKE=0 IMPLIES POLYNOMIALS ARE A FUNCTION OF TIME.
C
C     NSTRKE=1 IMPILES POLYNOMIALS ARE A FUNCTION OF STROKE.
C
C     IN EITHER CASE AN EJECTOR STROKE MUST BE SPECIFIED.
C
C     2. IF NSTRKE=0 EITHER TIME EXCEEDING THE SPECIFIED END OF EJECTOR
C     TIME OR THE STROKE EXCEEDING THE SPECIFIED LENGTH WILL CAUSE
C     THE EJECTOR FORCES + MOMENTS TO BE SET TO ZERO. SO IF TIME IS TO
C     BE THE SOLE CRITERION FOR EJECTOR TERMINATION, SET STROKE(I)
C     EQUAL TO A LARGE NUMBER.
C
C     3. IF NSTRKE=1 ONLY THE EJECTOR STROKE EXCEEDING THE
C     SPECIFICATION WILL CAUSE THE EJECTOR FORCES + MOMENTS TO BE SET
C     TO ZERO.
C
      COMMON /JECTOR/ NJECTR,FXS,FYS,FZS,RMX,RMY,RMZ
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSBY(51),FUSAZ(51)
     *   ,XJ(51),PHIK(33)
      COMMON /BOPTNS/ IOPTS(50)
      DIMENSION DC(3,3),VAR(12)
      DIMENSION XYZEI(2,3)
      DIMENSION STROKE(2),NEPOLY(2),TEEND(2,5),AKE(10,6)
      DIMENSION XE(2),THETAE(2)
      DIMENSION XEND(7),COEF(7,7)
      EQUIVALENCE (IOPTS(10),NFUSOR)
      LOGICAL ELLSTR
C
      RADELL(PHI,AZ,BY) = 1.0/SQRT((COS(PHI)/BY)**2+(SIN(PHI)/AZ)**2)
C
  702 FORMAT(2I5)
  711 FORMAT(I5,4F10.0)
  712 FORMAT(6F10.0)
  713 FORMAT(//10X,25HEJECTOR FORCE DESCRIPTION
     1   /15X,31HNUMBER OF EJECTOR FEET (NFEET=),I3)
  714 FORMAT(15X,44HEJECTOR FORCE IS FUNCTION OF TIME (NSTRKE=0))
  715 FORMAT(15X,46HEJECTOR FORCE IS FUNCTION OF STROKE (NSTRKE=1))
  716 FORMAT(/10X,12HEJECTOR FOOT,I5,27H,  LOCATION AND POLYNOMIALS
     1   /15X,35HNUMBER OF POLYNOMIALS      (NEPOLY=),I8
     2   /15X,35HAXIAL LOCATION AFT OF XMOM     (XE=),F8.4,5H FEET
     3   /15X,35HROLL ORIENTATION          (THETAE=),F8.2,8H DEGREES
     4   /15X,35HSTROKE LENGTH              (STRKE=),F8.4,5H FEET)
  717 FORMAT(10X,44HPOLYNOMIALS SPECIFYING EJECTOR FORCE HISTORY
     1   /15X,27HTIME/LENGTH OF EACH SECTION
```

Figure C-1(p)

```
     2   /17X,14HSECTION  TEENO,5(/15X,I5,G12.5))
718   FORMAT(10X,46HCOEFFICIENTS OF POLYNOMIALS DESCRIBING SECTION
     1   /10X,12HFOOT SECTION,3X,2HC0,11X,2HC1,11X,2HC2,11X,2HC3
     2   ,11X,2HC4,11X,2HC5)
719   FORMAT (10X,I3,I7,6G13.6)
C
C     IF HERE FIRST TIME NJECTR=1, FALL THRU AND READ EJECTOR
C     DATA
C
C     IF HERE SECOND TIME NJECTR=2, GO CALCULATE EJECTOR FOOT
C     LOCATIONS PRIOR TO SEPARATION.
C
C     IF NJECTR=3 GO CALCULATE EJECTOR FORCES + MOMENTS.
C
      IF (NJECTR.EQ.2) GO TO 120
      IF (NJECTR.EQ.3) GO TO 180
C
C     READ NFEET=NUMBER OF EJECTOR FEET
C     NSTRKE= SWITCH FOR STROKE OR TIME TO BE INDEPENDENT
C     VARIABLE
C
      READ(5,702) NFEET,NSTRKE
      WRITE(6,713) NFEET
      IF (NSTRKE.EQ.0) WRITE(6,714)
      IF (NSTRKE.EQ.1) WRITE(6,715)
C
      DO 100 I=1,NFEET
C
C     READ INFORMATION ON ITH FOOT
C
C     NEPOLY(I)= NUMBER OF POLYNOMIALS SPECIFYING THE EJECTOR FORCE.
C
C     XE(I)= THE DISTANCE TO THE ITH FOOT. SAME SIGN CONVENTION AS XS
C     AXIS IN FIGURE 26 OF AFFDL-TR-74-130 VOLUME II, BUT THE ORIGIN
C     HAS BEEN SHIFTED TO THE STORE CG,I.E. XE(I) IS NEGATIVE IF
C     THE FOOT IS LOCATED BETWEEN NOSE AND CG OF THE STORE.
C     (UNITS FEET)
C
C     THETAE(I)= ANGLE BETWEEN THE XS-ZS PLANE (VERTICAL) AND
C     THE LINE OF ACTION OF THE ITH EJECTOR FOOT. POSITIVE
C     COUNTERCLOCKWISE AS VIEWED FROM THE REAR OF THE STORE.
C     (UNITS DEGREES)
C
C     STROKE(I)= LENGTH OF ITH EJECTOR STROKE (UNITS FEET).
C
      READ(5,711) NEPOLY(I),XE(I),THETAE(I),STROKE(I)
      WRITE(6,716) I,NEPOLY(I),XE(I),THETAE(I),STROKE(I)
      THETAE(I)=THETAE(I)*DTOR
      READ(5,712) (TEENO(I,J),J=1,5)
      NEP = NEPOLY(I)
100   WRITE(6,717) (J,TEENO(I,J),J=1,NEP)
C
C     READ COEFFICIENTS OF 5TH DEGREE POLYNOMIAL
C
      WRITE(6,718)
      N=0
      DO 110 I=1,NFEET
      NEP = NEPOLY(I)
      DO 110 K=1,NEP
      N = N+1
      READ(5,712) (AKE(N,J),J=1,6)
110   WRITE(6,719) I,K,(AKE(N,J),J=1,6)
      NJECTR=2
      RETURN
120   NJECTR=3
C
C     CALCULATE EJECTOR FOOT LOCATION IN CARRIAGE
C     POSITION
C
      DO 170 I=1,NFEET
      XYZEI(I,1)=VAR(7)-XE(I)*COS(... (11))
      IF (ELLSTR) GO TO 130
C             USE CIRCULAR ...
      XL=(XMOM+XE(I))/STLGTH
      CALL SHAPE(XL,NPOLY,XENO,COEF,R,DRDX)
      R=R*STLGTH
      GO TO 160
C             USE RADIUS AT SUFACE OF ELLIPSE
C             FIND BY AND AZ, ELLIPTIC SEMI-AXES, AT XL
130   XL = XMOM+XE(I)
      DO 140 K=2,NFUSOR
      IF (XL.LE.XFUS(K)) GO TO 150
140   CONTINUE
      K = NFUSOR
150   DX = XFUS(K)-XFUS(K-1)
      IF (DX.NE.0.0) DX=(XL-XFUS(K-1))/DX
      AZ = (FUSAZ(K)-FUSAZ(K-1))*DX+FUSAZ(K-1)
      BY = (FUSBY(K)-FUSBY(K-1))*DX+FUSBY(K-1)
      PHI = THETAE(I)+VAR(12)
      R = RADELL(PHI,AZ,BY)
160   XYZEI(I,2)=VAR(8)-R*SIN(THETAE(I))
170   XYZEI(I,3)=VAR(9)-R*COS(THETAE(I))+XE(I)*SIN(VAR(11))
      RETURN
C
C     DEFAULT FORCE FUNCTION OF TIME
C
180   T=TIME
C
C     CALCULATE ADDITIONAL POINT ON STORE CENTERLINE
C     IN INERTIAL COORDINATE SYSTEM
C
      CALL STTOIN(1.0,0.0,0.0,XI,YI,ZI,DC)
      X1=VAR(7)
      X2=VAR(7)+XI
      Y1=VAR(8)
      Y2=VAR(8)+YI
      Z1=VAR(9)
      Z2=VAR(9)+ZI
C
C     INITIALIZE FORCES + MOMENTS
C      -
      FXS=0.0
      FYS=0.0
      FZS=0.0
      RMX=0.0
      RMY=0.0
      RMZ=0.0
C
C     LOOP 280 CALCULATES EJECTOR FORCES + MOMENTS
C
      DO 280 I=1,NFEET
C
C     CALCULATE LOCAL STORE RADIUS
C
      XL=XYZEI(I,1)-VAR(7)
      CALL INTOST(XL,0.0,0.0,XS,YS,ZS,DC)
      IF (ELLSTR) GO TO 190
C             USE CIRCULAR BODY RADIUS
      XL=(XMOM-XS)/STLGTH
      CALL SHAPE(XL,NPOLY,XENO,COEF,R,DRDX)
      R=R*STLGTH
      GO TO 220
C             USE RADIUS AT SUFACE OF ELLIPSE
C             FIND BY AND AZ, ELLIPTIC SEMI-AXES, AT XL
190   XL = XMOM+XE(I)
      DO 200 K=2,NFUSOR
      IF (XL.LE.XFUS(K)) GO TO 210
200   CONTINUE
      K = NFUSOR
210   DX = XFUS(K)-XFUS(K-1)
      IF (DX.NE.0.0) DX=(XL-XFUS(K-1))/DX
      AZ = (FUSAZ(K)-FUSAZ(K-1))*DX+FUSAZ(K-1)
      BY = (FUSBY(K)-FUSBY(K-1))*DX+FUSBY(K-1)
      PHI = THETAE(I)+VAR(12)
      R = RADELL(PHI,AZ,BY)
220   CONTINUE
C
C     CALCULATE INTERSECTION OF PLANE OF ACTION
```

Figure C-

```
C     OF EJECTOR AND STORE CENTERLINE
C
      A=Y1+((Y2-Y1)/(X2-X1))*(XYZEI(I,1)-X1)
      B=Z1+((Z2-Z1)/(X2-X1))*(XYZEI(I,1)-X1)
C
C     PROJECT R INTO PLANE OF ACTION
C
      C=R*COS(VAR(10))
      D=R*COS(VAR(11))
C
C     SM = SLOPE OF EJECTOR LINE OF ACTION
C
      IF(ABS(THETAE(I)).GT.0.00001) GO TO 230
      Y=XYZEI(I,2)
      Z=B-D
      GO TO 240
230   SM=TAN((90.*DTOR)-THETAE(I))
      E=-SM*XYZEI(I,2)+XYZEI(I,3)
      B=E-B
C
C     SOLVE FOR INTERSECTION OF EJECTOR FOOT +
C     STORE BODY
C
      A1=D**2+C**2*SM**2
      B1=2.0*(-A*D**2+C**2*SM*B)
      C1=D**2*A**2+C**2*(B**2-D**2)
      SQR=B1**2-4.0*A1*C1
      IF(SQR.LT.0.0) GO TO 280
      YR1=(-B1+SQRT(SQR))/(2.0*A1)
      YR2=(-B1-SQRT(SQR))/(2.0*A1)
      Y=YR2
      IF(THETAE(I).LT.0.0) Y=YR1
      Z=SM*Y-SM*XYZEI(I,2)+XYZEI(I,3)
C
C     CHECK EJECTOR STROKE
C
240   D=SQRT((XYZEI(I,2)-Y)**2+(XYZEI(I,3)-Z)**2)
C
C     IS STROKE GREATER THAN SPECIFICATION
C     IF SO SKIP TO END OF LOOP
C
      IF(D.GT.STROKE(I)) GO TO 280
C
C     IS STROKE TO BE INDEPENDENT VARIABLE
C
      IF(NSTRKE.EQ.1) T=D
C
C     CALCULATE EJECTOR FORCE
C
      NN=NEPOLY(I)
      DO 250 J=1,NN
      K=J
      IF(T.LE.TEEND(I,J)) GO TO 260
250   CONTINUE
C
C     IF INDEPENDENT VARIABLE IS GREATER THAN
C     SPECIFICATION SKIP TO END OF LOOP
C
      GO TO 280
260   J=K
      IF(I.NE.1) J=J+NEPOLY(1)
      F=AKE(J,1)+T*(AKE(J,2)+T*(AKE(J,3)+
     1T*(AKE(J,4)+T*(AKE(J,5)+
     1T*(AKE(J,6))))))
C
C     PROJECT INTO INERTIAL Y-Z PLANE
C
      FY=F*SIN(THETAE(I))
      FZ=F*COS(THETAE(I))
C
C     PROJECT FORCES INTO STORE COORDINATE SYSTEM
C
      CALL INTOST(0.0,FY,FZ,FXSS,FYSS,FZSS,DC)
      FXS=FXSS+FXS
      FYS=FYSS+FYS
      FZS=FZSS+FZS
C
C     FORCES WILL ADD INTO EQUATIONS OF MOTION
C
      X=XYZEI(I,1)
C
C     SHIFT ORIGIN TO STORE CG
C
      X=X-VAR(7)
      Y=Y-VAR(8)
      Z=Z-VAR(9)
      CALL INTOST(X,Y,Z,XS,YS,ZS,DC)
C
C     FXS CAUSES PITCHING + YAWING MOMENTS
C
      RMZ=-FXSS*YS+RMZ
      RMY=FXSS*ZS+RMY
C
C     FYS    CAUSES YAWING + ROLLING MOMENTS
C
      RMZ=FYSS*XS+RMZ
      RMX=-FYSS*ZS+RMX
C
C     FZS  CAUSES ROLLING + PITCHING MOMENTS
C
      RMX=FZSS*YS+RMX
      RMY=-FZSS*XS+RMY
280   CONTINUE
      RETURN
      END


      SUBROUTINE ELI1(RES,X,CK)
C
C     ..................................................................
C
C        SUBROUTINE ELI1
C
C        PURPOSE
C           COMPUTES THE ELLIPTIC INTEGRAL OF FIRST KIND
C
C        USAGE
C           CALL ELI1(RES,X,CK)
C
C        DESCRIPTION OF PARAMETERS
C           RES    - RESULT VALUE
C           X      - UPPER INTEGRATION BOUND (ARGUMENT OF ELLIPTIC
C                    INTEGRAL OF FIRST KIND)
C           CK     - COMPLEMENTARY MODULUS
C
C        REMARKS
C           MODULUS K = SQRT(1.-CK*CK).
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           RES=INTEGRAL(1/SQRT((1+T*T)*(1+(CK*T)**2)), SUMMED
C           OVER T FROM 0 TO X).
C           EQUIVALENT ARE THE DEFINITIONS
C           RES=INTEGRAL(1/(COS(T)*SQRT(1+(CK*TAN(T))**2)), SUMMED
C           OVER T FROM 0 TO ATAN(X)),
C           RES=INTEGRAL(1/SQRT(1-(K*SIN(T))**2), SUMMED OVER
C           T FROM 0 TO ATAN(X)).
C           EVALUATION
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R. BULIRSCH, NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS AND
C                  ELLIPTIC FUNCTIONS.
C                  HANDBOOK SERIES OF SPECIAL FUNCTIONS
```

Figure C-1(r)

```
C                  NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ..................................................................
C
C
      IF(X)2,1,2
    1 RES=0.
      RETURN
    2 IF(CK)4,3,4
    3 RES=ALOG(ABS(X)+SQRT(1.+X*X))
      GOTO 13
    4 ANGLE=ABS(1./X)
      GEO=ABS(CK)
      ARI=1.
      PIM=0.
    5 SQGEO=ARI*GEO
      AARI=ARI
      ARI=GEO+ARI
      ANGLE=-SQGEO/ANGLE+ANGLE
      SQGEO=SQRT(SQGEO)
      IF(ANGLE)7,6,7
C     REPLACE 0 BY SMALL VALUE
    6 ANGLE=SQGEO*1.E-8
    7 TEST=AARI*1.E-4
      IF(ABS(AARI-GEO)-TEST)10,10,8
    8 GEO=SQGEO+SQGEO
      PIM=PIM+PIM
      IF(ANGLE)9,5,5
    9 PIM=PIM+3.1415927
      GOTO 5
   10 IF(ANGLE)11,12,12
   11 PIM=PIM+3.1415927
   12 RES=(ATAN(ARI/ANGLE)+PIM)/ARI
   13 IF(X)14,15,15
   14 RES=-RES
   15 RETURN
      END


      SUBROUTINE ELI2(R,X,CK,A,B)
C
C     ..................................................................
C
C        SUBROUTINE ELI2
C
C        PURPOSE
C           COMPUTES THE GENERALIZED ELLIPTIC INTEGRAL OF SECOND KIND
C
C        USAGE
C           CALL ELI2(R,X,CK,A,B)
C
C        DESCRIPTION OF PARAMETERS
C           R      - RESULT VALUE
C           X      - UPPER INTEGRATION BOUND (ARGUMENT OF ELLIPTIC
C                    INTEGRAL OF SECOND KIND)
C           CK     - COMPLEMENTARY MODULUS
C           A      - CONSTANT TERM IN NUMERATOR
C           B      - QUADRATIC TERM IN NUMERATOR
C
C        REMARKS
C           MODULUS K = SQRT(1.-CK*CK).
C           SPECIAL CASES OF THE GENERALIZED ELLIPTIC INTEGRAL OF
C           SECOND KIND ARE
C           F(ATAN(X),K) OBTAINED WITH A=1., B=1.
C           E(ATAN(X),K) OBTAINED WITH A=1., B=CK*CK.
C           B(ATAN(X),K) OBTAINED WITH A=1., B=0.
C           D(ATAN(X),K) OBTAINED WITH A=0., B=1.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           R=INTEGRAL((A+B*T*T)/(SQRT((1+T*T)*(1+(CK*T)**2))*(1+T*T)),
C                SUMMED OVER T FROM 0 TO X).
C           EQUIVALENT IS THE DEFINITION
C           R=INTEGRAL((A+(B-A)*(SIN(T))**2)/SQRT(1-(K*SIN(T))**2),
C                SUMMED OVER T FROM 0 TO ATAN(X)).
C           EVALUATION
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R. BULIRSCH, NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS AND
C                ELLIPTIC FUNCTIONS
C                HANDBOOK SERIES OF SPECIAL FUNCTIONS
C                NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ..................................................................
C
C        TEST ARGUMENT
      IF(X)2,1,2
    1 R=0.
      RETURN
C        TEST MODULUS
    2 C=0.
      D=0.5
      IF(CK)7,3,7
    3 R=SQRT(1.+X*X)
      R=(A-B)*ABS(X)/R+B*ALOG(ABS(X)+R)
C        TEST SIGN OF ARGUMENT
    4 R=R+C*(A-B)
      IF(X)5,6,6
    5 R=-R
    6 RETURN
C        INITIALIZATION
    7 AN=(B+A)*0.5
      AA=A
      R=B
      ANG=ABS(1./X)
      PIM=0.
      ISI=0
      ARI=1.
      GEO=ABS(CK)
C        LANDEN TRANSFORMATION
    8 R=AA*GEO+R
      SGEO=ARI*GEO
      AA=AN
      AARI=ARI
C        ARITHMETIC MEAN
      ARI=GEO+ARI
C        SUM OF SINE VALUES
      AN=(R/ARI+AA)*0.5
      AANG=ABS(ANG)
      ANG=-SGEO/ANG+ANG
      PIMA=PIM
      IF(ANG)10,9,11
    9 ANG=-1.E-8*AANG
   10 PIM=PIM+3.1415927
      ISI=ISI+1
   11 AANG=ARI*ARI+ANG*ANG
      P=D/SQRT(AANG)
      IF(ISI-4)13,12,12
   12 ISI=ISI-4
   13 IF(ISI-2)15,14,14
   14 P=-P
   15 C=C+P
      D=D*(AARI-GEO)*0.5/ARI
      IF(ABS(AARI-GEO)-1.E-4*AARI)17,17,16
   16 SGEO=SQRT(SGEO)
C        GEOMETRIC MEAN
      GEO=SGEO+SGEO
      PIM=PIM+PIMA
      ISI=ISI+ISI
      GOTO 8
C        ACCURACY WAS SUFFICIENT
   17 R=(ATAN(ARI/ANG)+PIM)*AN/ARI
      C=C+D*ANG/AANG
```

Figure C-1(s)

```
      GOTO 4
      END


      SUBROUTINE ELLCPT (YCP,ZCP,RA,RB)
C
C  EXTRAPOLATION OF CONTROL POINTS TO LOCAL SURFACE OF ELLIPSE.
C  PROJECTION IS MADE ALONG RADIAL LINE FROM ORIGIN THRU CONTROL POINT
C
      RCPT = SQRT(YCP*YCP+ZCP*ZCP)
      SINTH = ZCP/RCPT
      COSTH = YCP/RCPT
      RBODY = SQRT(1.0/((SINTH/RA)**2+(COSTH/RB)**2))
      IF (RCPT.GT.RBODY) GO TO 100
C
C  REDEFINE CONTROL POINT TO SURFACE OF ELLIPSE
      YCP = RBODY*COSTH
      ZCP = RBODY*SINTH
100   RETURN
      END


      SUBROUTINE ELLSHP (X,BY,AZ,BYP,AZP)
C
C  ROUTINE TO COMPUTE HORIZONTAL AND VERTICAL SEMI-AXES (BY,AZ)
C  AND THEIR DERIVATIVES WITH AXIAL STATION (BYP=DBY/DX, AZP=DAZ/DX)
C  FOR AN ELLIPSE  BY TABLE LOOKUP.
C
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSBY(51)
     *    ,FUSAZ(51),XJ(51),PHIK(33)
      COMMON /BOPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BOOL,IZ1(12)
      COMMON /COM9  / IGROW
      COMMON /VEL   / VXAV,WXAV,USTOR,NOUT,NV,NS,NF
C
C  FIND INDICES BRACKETING INPUT VALUE OF X.
C
      IGROW = 0
      IJ = 0
      DO 100 J=1,NFUS
      IJ = IJ+1
      NX = NFORX(J)
      DO 100 I=2,NX
      IJ = IJ+1
      IF (X.LE.XFUS(IJ)) GO TO 120
100   CONTINUE
C
C  INTERPOLATE FOR AZ,BY AT IJ
C
120   IM1 = IJ-1
      X1 = XFUS(IJ)
      X0 = XFUS(IM1)
      DXI = X1-X0
      IF (DXI.NE.0.) DXI=1.0/DXI
      BYP = (FUSBY(IJ)-FUSBY(IM1))*DXI
      AZP = (FUSAZ(IJ)-FUSAZ(IM1))*DXI
      AZ  = FUSAZ(IM1)+AZP*(X-X0)
      BY  = FUSBY(IM1)+BYP*(X-X0)
      IF (ABS(BYP).GT.0.002 .OR. ABS(AZP).GT.0.002) IGROW=1
      RETURN
      END


      SUBROUTINE ELRFLB (ALPHAC,PHIR,VAR12)
C
C  ROUTINE TO COMPUTE THE IMAGE ELLIPTIC STORE LOCATION RELATIVE TO THE
C  FUSELAGE.  THE FUSELAGE BODY MAY BE CIRCULAR OR NON-CIRCULAR.
C
      LOGICAL TEST
      DIMENSION  IA(1)
      DIMENSION SX(30),SR(30)
      COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,DBETA
     *    ,EALPHA,CNU0,CNU2,XSHLDR,SHK(3),XSHK(100),RSHK(100)
      COMMON /BSWINT/ IP,XPTIP,YPTIP,ZPTIP,DELTIP,DRDXS,YBS,ZBS,BTNOSN
      COMMON /CONFIG/ NFU,NPY,NSTRS,NRACK
      COMMON /CONSTS/ PI,DTOR
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,UC(3,3)
     *    ,CDC,XMOM,ESTHMA,DX,DELX,VSTORE,ESTLGC
      COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101),
     *   FRMAX,FLTHC,NFPOLY,FXEND(7),FCOEF(7,7)
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1   ,IDEJ,IDF,ID2
      COMMON /STORIC/ RTNOSC,RSSC
      COMMON /STORIF/ IMFSTR,BTNOSF,RSS,YBN,ZBN,SFAC
      COMMON          A(12000)
      EQUIVALENCE (IA(1),A(1))
C
C  STORE NOSE LOCATION IN FUSELAGE COORDINATES
      CALL STTOIN (XMOM,0.,0.,X1,ETA,ZETA,DC)
      XBN = VAR(7)+X1
      YBN = VAR(8)+ETA
      ZBN = VAR(9)+ZETA
      RRN = SQRT(YBN*YBN+ZBN*ZBN)
C
C  MERIDIAN ANGLE BETWEEN STORE CENTERLINE AND FUSELAGE CENTERLINE
      PHISF = ATAN2(-YBN,-ZBN)
C
C  ANGLE OF FUSELAGE WITH STORE AXIS
C
      PHISHK = PHISF+VAR12*DTOR
C
C  GENERATE SHOCK TRAVERSE FOR X VS R IN FUSELAGE COORDINATES
C
      ALPR = ALPHAC*(1.-EALPHA)*DTOR
      SVAR = SIN(PHISHK)
      CVAR = COS(PHISHK)
      CALP = COS(ALPR)
      SALP = SIN(ALPR)
      NS = NSHK(1)
      DO 100 I=1,NS
      RS = RSHK(I)
      YS = RS*SVAR
      ZS = -RS*CVAR
      CALL XVSRT(XS,YS,ZS,NSHK,PHIS,NSHOCK,XSHK,RSHK,ALPR,PHIR,.TRUE.)
      SX(I) = XS*CALP-RS*SALP
      SR(I) = XS*SALP+RS*CALP
100   CONTINUE
C
C  TEST FOR CIRCULAR OR NON-CIRCULAR FUSELAGE OPTION
C
      XZ = -XBN
      IF (NFU.EQ.2) GO TO 130
C
C  CALCULATE ELLIPTIC STORE INTERSECTION WITH CIRCULAR FUSELAGE ---------
C
      CALL SWINT (XZ,RBN,NS,SX,SR,FXEND,FCOEF,NFPOLY,FLTHC,
     *   FRMAX,DRDXS,DRDXB,XBS,RBS,IMFSTR)
      IF (IMFSTR.EQ.0) RETURN
      XBS = -XBS
C
C  CALCULATE SHOCK WAVE REFLECTION ANGLE
      THETAR = ATAN(DRDXS)+2.*ATAN(DRDXB)
      DRDXR = -TAN(THETAR)
      THETAR = THETAR/DTOR
      GO TO 140
C
C  CALCULATE ELLIPTIC STORE SHOCK INTERSECTION WITH NON-CIRCULAR FUS ---
C  INDICES OF FUSELAGE ARRAYS IN BLANK COMMON
C
130   ID0 = IA(IDF+74)
```

Figure C-1(t)

```
      IA0 = IA(IDF+73)
      IXPT = IA(IDO+42)+IA0
      IYPT = IA(IDO+43)+IA0
      IZPT = IA(IDO+44)+IA0
      ITH  = IA(IDO+45)+IA0
      IDEL = IA(IDO+46)+IA0
      IXC  = IA(IDO+23)+IA0
      IKR  = IDO+13
      IKX  = IDO+18
      NFUS = IA(IDO+12)
      BLN = A(IDO+180)
C
C  COMPUTE XBS AND RBS OF SHOCK INTERSECTION WITH FUSELAGE
C
      CALL SWINTE (XZ,YBN,ZBN,NS,SX,SR,BLN,XBS,RBS,NFUS,IMFSTR
     *   ,IA(IKX),IA(IKR),A(IXPT),A(IYPT),A(IZPT),A(ITH),A(IDEL),A(IXC))
      IF (IMFSTR.EQ.0) RETURN
      XBS = -XBS
C
C  COMPUTE REFLECTION ANGLE TANGENT
C
      TAN2D = TAN(2.*DELTIP)
      DRDXR = -(TAN2D-DRDXS)/(1.+TAN2D*DRDXS)
      THETAR = ATAN(DRDXR)
C
C  LOCATE STORE CG IN BODY SYSTEM AND CALCULATE RADIAL DISTANCE
C  TO SURFACE OF FUSELAGE ASSUMING CG LIES IN PLANE CONTAINING
C  SHOCK AND PANEL CONTROL POINT
C
140   XBCG = VAR(7)
      YBCG = VAR(8)
      ZBCG = VAR(9)
      RCG = SQRT(YBCG*YBCG+ZBCG*ZBCG)
C
C  DETERMINE INTERSECTION OF REFLECTED SHOCK AND LINE FROM STORE
C  NOSE THROUGH CG.
C    EQUATION OF STORE AXIS$   (R-RBN)/(RCG-RBN) = (X-XBN)/(XBCG-XBN)
      TALF = (RCG-RBN)/(XBCG-XBN)
      RAX = RBN-XBN*TALF
C
C    EQUATION OF REFLECTED SHOCK $   (R-RBS) = DRDXR (X-XBS)
      RRS = RBS-XBS*DRDXR
      TEST = DRDXR.GT.TALF
      IF (TEST) IMFSTR=0
      IF (TEST) GO TO 999
      TEST = ABS(DRDXR).GT.100.
      IF (TEST) XBR=XBS
      IF (TEST) GO TO 150
C
C  INTERSECTION$   RAX+X*TALF = RRS+X*TAN(THETAR)
      XBR = (RRS-RAX)/(TALF-DRDXR)
150   RBR = RAX+XBR*TALF
C
C  SEE IF INTERSECTION WITH STORE AXIS IS AFT OF STORE BASE
      ALFR = ATAN(-TALF)
      XSR = (XBN-XBR)*COS(ALFR)+(RBR-RBN)*SIN(ALFR)
      TEST = XSR.GE.ESTLGC
      IF (TEST) IMFSTR=0
      IF (TEST) GO TO 999
      IMFSTR = 1
C
C  CALCULATE VALUE OF BETA TO USE FOR IMAGE STORE NOSE
C
      RSSC = RBN
      RSS = RSSC-RBS*RBS/RSSC
      BTNOSC = XSR/RSSC
      BTNOSF = XSR/RSS
      BTNOSN = XSR/(2.*(RBN-RBS))
      SFAC = RBS*RBS/(RSSC*RSSC)
999   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELRFLW (ALPHAC,PHIR,VAR12)
C
C  ROUTINE TO CALCULATE SHOCK SHAPE BETWEEN STORE AND WING SURFACE.
C  SHAPE IS USE TO DETERMINE IMAGE STORE LOCATION FOR ELLIPTIC STORES.
C  NO WING DIHEDRAL  EFFECTS ARE CONSIDERED IN SHAPE DETERMINATION.
C
      DIMENSION SX(30),SR(30)
      COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,DBETA
     *   ,EALPHA,CNU0,CNU2,XSHLDR,SHK(3),XSHK(100),RSHK(100)
      COMMON /CONSTS/ PI,DTOR
      IF (NSHOCK.LE.0) WRITE(6,700) NSHOCK
700   FORMAT (9H  NSHOCK=,I5,80X,12H** ELREFL **)
C
C  DEFINE SHOCK SHAPE AT VAR12
C
      PHI = (180.+VAR12)*DTOR
      CVAR = COS(PHI)
      SVAR = SIN(PHI)
      NS = NSHK(1)
      ALPD = ALPHAC*(1.-EALPHA)
      ALPR = ALPD*DTOR
      SALP = SIN(ALPR)
      CALP = COS(ALPR)
C
C  USE FIRST TRAVERSE TO DEFINE R FOR ROTATION
C
      DO 100 I=1,NS
      RS = RSHK(I)
      YS = RS*SVAR
      ZS = RS*CVAR
      CALL XVSRT(XS,YS,ZS,NSHK,PHIS,NSHOCK,XSHK,RSHK,ALPR,PHIR,.TRUE.)
      SX(I) = XS*CALP-RS*SALP
      SR(I) = XS*SALP+RS*CALP
100   CONTINUE
      CALL REFSHK (NS,SX,SR)
      RETURN
      END
```

```
      SUBROUTINE EXPAND (BYP,AZP,X,Y,VS,WS)
C
C     THIS SUBROUTINE CALCULATES VELOCITIES INDUCED BY EXPANDING OR
C     CONTRACTING ELLIPTICAL CROSS SECTION.
C     ELLIPSE IS ASSUMED TO HAVE INDEPENDENTLY VARYING ELLIPTIC AXES
C     AZ AND BY.
C
      COMMON/COM1/A2,B2,R2
      COMMON/BLK1/BY,AZ,R,S,PI,CI
      COMPLEX VE,Z,ZI,CI
      SP=.5*(BY*AZP+AZ*BYP)
      Z=CMPLX(X,Y)
      ZI=Z*Z
      AY=1.0
      IF(AIMAG(Z).LT.0.0) AY=-1.0
      AYZ=1.0
      IF(AIMAG(ZI).LT.0.0) AYZ=-1.0
      Z=CSQRT(ZI-A2+B2)*AY*AYZ
      IF((ABS(AIMAG(Z)).LE.0.0).AND.(REAL(Z).LT.0.0))
     1      Z=CMPLX(-REAL(Z),AIMAG(Z))
      VE=SP/Z
      VS=REAL(VE)
      WS=-AIMAG(VE)
      RETURN
      END
```

```
      SUBROUTINE F(XO,PX,N,WK)
C
```

Figure C-1(u)

```
C     THIS SUBROUTINE IS CALLED BY DASCRU TO CALCULATE CROSS FLOW
C     PLANE VELOCITIES
C
      COMPLEX CI
C
      COMMON/BLK1/BY,AZ,R,S,PI,CI
      COMMON/COM1/A2,B2,R2
      COMMON/COM2/SIG2,H2
      COMMON/COM9/IGROW
      COMMON /TEVRT / G(20),XV(20),YV(20)
      COMMON /VEL   / VXAV,WXAV,USTOR,NOUT,NV,NS,NF
C
C
      DIMENSION XO(1),WK(1),V(20),W(20),VV(20),WW(20)
C
C
     CI=CMPLX(0.,1.)
     PI=3.14159265
     DO 90 I=1,NV
     J=2*I-1
     K=J+1
     XV(I)=XO(J)
  90 YV(I)=XO(K)
     CALL ELLSHP (PX,BY,AZ,BYP,AZP)
C
C NOTES COORDINATE XV(I) IS THE Y-COORDINATE IN THE CROSS FLOW PLANE.
C         COORDINATE YV(I) IS THE Z-COORDINATE IN THE CROSS FLOW PLANE.
C
C  .. COMPUTE VELOCITIES IN THE CROSSFLOW PLANE ...
C     V(I) IS IN THE Y-DIRECTION.
C     W(I) IS IN THE Z-DIRECTION.
C
C  SPECIAL CASE. NO FINS ALONG BODY LENGTH UNDER CONSIDERATION. S=0
C
C
     S =BY
     A2=BY*BY
     B2=AZ*AZ
     APB=AZ+BY
     APB2=APB*APB
     R1=S+SQRT(S*S-A2+B2)
     R=0.25*(R1+APB2/R1)
     H=R+R
     H2=H*H
     SIG=0.5*APB
     SIG2=SIG*SIG
     R2=R*R
     DO 100 I=1,NV
     X=XV(I)
     Y=YV(I)
     VS=0.
     WS=0.
C
C    COMPUTE VELOCITIES IN CROSS FLOW PLANE DUE TO ANGLE OF PITCH AND
C    ANGLE OF SIDESLIP.
C
     CALL PITROL(PX,VXAV,WXAV,X,Y,VS,WS)
     V(I)=VS
     W(I)=WS
C
C    COMPUTE VELOCITIES IN CROSSFLOW PLANE DUE TO EXPANDING BODY
C
     VS=0.0
     WS=0.0
     IF (IGROW.GT.0) CALL EXPAND(BYP,AZP,X,Y,VS,WS)
     V(I)=V(I)+VS*USTOR
     W(I)=W(I)+WS*USTOR
 100 CONTINUE
C
C    CALCULATE VELOCITIES INDUCED BY ALL VORTICES ON ONE VORTEX IN THE
C    PRESENCE OF AN ELLIPTICAL BODY-MONOPLANE WING.
C
     CALL VOTEX(NV,XV,YV,G,VV,WW)
     DO 101 I=1,NV
     V(I)=V(I)+VV(I)
     W(I)=W(I)+WW(I)
     J=2*I-1
     K=J+1
     WK(J)=V(I)/USTOR
     WK(K)=W(I)/USTOR
 101 CONTINUE
     RETURN
     END


      SUBROUTINE FLDAIC (XFP,YFP,ZFP,SVN,LSKP,UB,NFLD)
C
C  ROUTINE TO GENERATE INFLUENCE COEFFICIENTS DUE TO BODY SOURCE PANELS
C  AT FIELD CONTROL POINTS (XFP,YFP,ZFP)
C
      DIMENSION XFP(NFLD),YFP(NFLD),ZFP(NFLD),UB(3,1)
     *    ,SVN(NFLD),LSKP(NFLD)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NACIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *    ,IAR,IAN,IGR,IGH(7),IVP,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *    ,NAG,NAP,NAV,NAS,NASHW,NAFLD,IAO,IDO,ISKO,IZZ(4),NRING,IROW(50)
      COMMON          A(12000)
C
C  GENERATE VELOCITY COEFFICIENTS INDUCED BY PANELS IN EACH BODY
C  SEGMENT
C
      DO 150 IFU=1,KFUS
      JR = KRADX(IFU)-1
      IXCO = IXC(IFU)
      IYCO = IYC(IFU)
      IZCO = IZC(IFU)
      CALL FLDAC2 (A(IXPT),A(IYPT),A(IZPT),A(ITH),A(IDEL)
     *    ,XFP,YFP,ZFP,UB,SVN,LSKP,NFLD
     *    ,A(IXCO),A(IYCO),A(IZCO),KFORX(IFU),IXZSYM,IFU,JR)
 150  CONTINUE
      RETURN
      END


      SUBROUTINE FLDAC2 (XPT,YPT,ZPT,THET,DELTA,XFP,YFP,ZFP
     *    ,UB,SVN,LSKP,NFLD,XC,YC,ZC,KFUSOR,IXZSYM,IFU,JR)
C
C  ROUTINE TO COMPUTE THE INFLUENCE COEFFICIENTS FOR THE U,V,W VELOCITY
C  COMPONENTS INDUCED BY STORE BODY PANELS AT EXTERNAL FIELD POINTS
C
      COMMON /BLKPAN/ COST,SINT,XRTJ,YRTJ,ZRTJ,XCI,YCI,ZCI
     *    ,XPTI,YPTI,ZPTI,COSTI,SINTI,COSD,SIND,LZERO
      COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4)
     1    ,XI,YI,ZI,XJ,ZJ,BETAD,BETAL,SUBSON,SUPERS,J
      DIMENSION XPT(20),YPT(20),ZPT(20),THET(20),DELTA(20)
     *    ,XFP(NFLD),YFP(NFLD),ZFP(NFLD),SVN(20),UB(3,1)
     *    ,XC(KFUSOR),YC(KFUSOR,1),ZC(KFUSOR,1)
      LOGICAL LSKP(1)
      LOGICAL LZERO,SUBSON,SUPERS
      LZERO = .FALSE.
      BETAL = BETAD
C
      SINTI = 0.
      COSTI = 1.
      SIND  = 0.
      COSD  = 1.
      XCOR(1)=0.
      YCOR(1)=0.
      ZCOR(1)=0.
      XCOR(3)=0.
      DO 10 I=1,NFLD
 10   LSKP(I) = .FALSE.
C
```

Figure C-1(v)

```
C  JB IS STARTING OFFSET OF GB(J)
C  I IS THE INDEX OF THE FIELD POINT
C  J IS THE INDEX OF THE INFLUENCING PANEL
C
C             LOOP THROUGH NUMBER OF RINGS IN SEGMENT OF INFLUENCING
C             PANEL BLOCKS.  L - INDEX OF CROSS SECTION IN THIS SEGMENT
      IF (IFU.LE.1) JG=0
      DO 110 L=2,KFUSOR
C
   INITIALIZE SKIP LOGIC CHECK ON VELOCITY CONTRIBUTION
C  OF SUBSEQUENT PANELS OF A SINGLE RING
C
      DO 30 I=1,NFLD
30    SVN(I) = 0.
C
C  SETUP LOCAL GEOMETRY FOR INFLUENCING PANEL OF RING JB
      IJ = 0
      DO 60 N=1,JR
      J = JG+N
      XBTJ = XPT(J)
      YBTJ = YPT(J)
      ZBTJ = ZPT(J)
      TAND=TAN(DELTA(J))
      COST=COS(THET(J))
      SINT=SIN(THET(J))
      NP1 = N+1
      LM1 = L-1
      XC1=XC(LM1)
      YC1=YC(LM1,NP1)
      ZC1=ZC(LM1,NP1)
C
C  CALCULATION OF PANEL CORNER POINTS IN PANEL COORDINATE SYSTEM
C             K= 1       LM1= L-1        NP1= N+1
C                2            L               N+1
C                3            L-1             N
C                4            L               N
C
      XCOR(2)=XC(L)-XC1
      XCOR(4)=XCOR(2)
      DO 40 K=2,4
      LM1 = L
      NP1 = N+1
      IF (K.GE.3) NP1=N
      IF (K.EQ.3) LM1=L-1
      DELY=YC(LM1,NP1)-YC1
      DELZ=ZC(LM1,NP1)-ZC1
      YCOR(K)=DELY*COST+DELZ*SINT
      ZCOR(K)=DELZ*COST-DELY*SINT
40    CONTINUE
      CX=XCOR(2)
C             LOOP COLUMNWISE THROUGH ALL FIELD POINTS
      DO 50 I=1,NFLD
      IJ = IJ+1
      UB(1,IJ) = 0.0
      UB(2,IJ) = 0.0
      UB(3,IJ) = 0.0
      LZERO = .FALSE.
      IF (LSKP(I)) GO TO 50
      XPTI = XFP(I)
      YPTI = YFP(I)
      ZPTI = ZFP(I)
45    CALL PANVEL (UB(1,IJ),UB(2,IJ),UB(3,IJ),ANIJ,IXZSYM)
      SVN(I) = SVN(I)+UB(1,IJ)**2+UB(2,IJ)** 2+UB(3,IJ)**2
      IF (I.GT.1) LZERO=.TRUE.
50    CONTINUE
60    CONTINUE
C
C  SAVE COEFFICIENTS ON FIELD POINTS BY A GIVEN RING
C
      NB = 3*IJ
      CALL IOWRIT(10,UB,NB)
C
C  CHECK WHETHER CONTRIBUTION OF RING IS ZERO.
      IF (SUBSON) GO TO 110
      DO 70 I=1,NFLD
70    LSKP(I) = SVN(I).EQ.0.
110   JG = JG+JR
      RETURN
      END


      SUBROUTINE FLDUVW (UB,GB,U,V,W,NFLD,JROW,NRING)
C
C  ROUTINE TO COMPUTE U,V,W AT FIELD POINTS FROM COEFFICIENT MATRIX.
C  INFLUENCE COEFFICIENTS ARE STORED ON EXTERNAL FILE TAPE10.
C
      DIMENSION UB(3,1),U(NFLD),V(NFLD),W(NFLD),JROW(NRING),GB(1)
C
C  INITIALIZE U,V,W
C
      DO 100 I=1,NFLD
      U(I) = 0.0
      V(I) = 0.0
      W(I) = 0.0
100   CONTINUE
C
C
C  MATRIX MULTIPLICATION  U=UB(I)*GB BY RING
      JG = 0
      DO 140 JB=1,NRING
      JR = JROW(JB)
      NB = 3*NFLD*JR
      CALL IOREAD (10,UB,NB)
      IJ = 0
      DO 120 N=1,JR
      J = JG+N
      GBJ = GB(J)
      DO 120 I=1,NFLD
      IJ = IJ+1
      U(I) = U(I)+UB(1,IJ)*GBJ
      V(I) = V(I)+UB(2,IJ)*GBJ
      W(I) = W(I)+UB(3,IJ)*GBJ
120   CONTINUE
140   JG = JG+JR
      RETURN
      END


      SUBROUTINE FLDVEL (XFP,YFP,ZFP,SVN,LSKP,U,V,W,NFLD,ID,IG)
C
C  ROUTINE TO COMPUTE U,V,W VELOCITY COMPONENTS AT THE FIELD POINTS
C  XFP,YFP,ZFP.
C  ALL FIELD POINTS ARE ASSUMED TO HAVE THE SAME BTNOSE (I.E. LOCATED
C  AT SAME YS,ZS FROM BODY.)
C               ********** PROGRAM II VERSION **********
C
      COMMON /BOPTNS/ IOPTS(50)
      COMMON          A(12000)
      LOGICAL LSKP(NFLD),LA(1),INLETS,INLET
      DIMENSION XFP(NFLD),YFP(NFLD),ZFP(NFLD),U(NFLD),V(NFLD),W(NFLD)
     *    ,SVN(NFLD),ID(130)
      EQUIVALENCE (A(1),LA(1)),  (IOPTS(18),INLET)
      DATA IOLAST/-1/
      IF (IG.LT.1) IG=1
C
C  INITIALIZE VELOCITY COMPONENTS TO ZERO.
C
      DO 100 I=1,NFLD
      LSKP(I) = .FALSE.
      U(I) = 0.
      V(I) = 0.
100   W(I) = 0.
C             DEFINE INDEX OF FIRST LOCATION IN A-ARRAY OF PANEL
C             GEOMETRY AND STRENGTH
```

Figure C-1(w)

```
C             **** THE NEXT STATEMENT IS USED TO AVOID UNNECESSARY *****
C             **** CALCULATIONS. IT IS VALID ONLY IN PROGRAM 2 WHEN **
C             **** MULTIPLE CONFIGURATIONS ARE STACKED IN BLANK COMMON *
      IF (IDLAST.EQ.ID(73)) GO TO 120
      IAO = ID(73)
      IXPT = ID(42)+IAO
      IYPT = ID(43)+IAO
      IZPT = ID(44)+IAO
      ITH  = ID(45)+IAO
      IDEL = ID(46)+IAO
      IGB  = ID(IG+50)+IAO
      KFUS = ID(12)
      IXZSYM = ID(38)
      IDLAST = IAO
C
C  SUM INFLUENCES OF EACH BODY RING ON FIELD POINTS
C
120   JG = 0
      INLETS = INLET
C             IF SECOND STORE SHAPE OR FUSELAGE SET TO INLET OF CONFIG
      ID0 = ID(74)
      IF (IAO.GT.0) INLET=LA(ID0+160)
      DO 150 IFU=1,KFUS
      JR = ID(IFU+12)-1
      IXC = ID(IFU+22)+IAO
      IYC = ID(IFU+27)+IAO
      IZC = ID(IFU+32)+IAO
C
C             LOOP THROUGH NUMBER OF RINGS IN SEGMENT OF INFLUENCING
C             PANEL BLOCKS.  L - INDEX OF CROSS SECTION IN THIS SEGMENT
C
      KFUSOR = ID(IFU+17)
      DO 140 L=2,KFUSOR
      CALL FLDVL2 (A(IXPT),A(IYPT),A(IZPT),A(ITH),A(IDEL),A(IGB)
     * ,XFP,YFP,ZFP,U,V,W,SVN,LSKP,NFLD
     * ,A(IXC),A(IYC),A(IZC),KFUSOR,IXZSYM,JR,JG,L)
140   JG = JG+JR
150   CONTINUE
      INLET = INLETS
      RETURN
      END


      SUBROUTINE FLDVL2 (XPT,YPT,ZPT,THET,DELTA,GB,XFP,YFP,ZFP,U,V,W
     * ,SVN,LSKP,NFLD,XC,YC,ZC,KFUSOR,IXZSYM,JR,JG,L)
C
C  COMPUTE THE THREE COMPONENTS OF VELOCITY INDUCED AT SPECIFIED
C  CONTROL POINTS BY THE BODY PANELS OF A GIVEN BODY SEGMENT
C
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI
     * ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BLKPAN/ COST,SINT,XBTJ,YBTJ,ZBTJ,XC1,YC1,ZC1
     * ,XPTI,YPTI,ZPTI,COSTI,SINTI,COSD,SIND,LZERO
      COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4)
    1 ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,J
C
      DIMENSION XPT(20),YPT(20),ZPT(20),THET(20),DELTA(20)
     * ,GB(20),U(20),V(20),W(20),XFP(20),YFP(20),ZFP(20),SVN(20)
     * ,XC(KFUSOR),YC(KFUSOR,1),ZC(KFUSOR,1)
      LOGICAL LSKP(1)
      LOGICAL LZERO,SUBSON,SUPERS
      LOGICAL INLTST,NOINLT,OPEN
      EXTERNAL INLTST
      LZERO = .FALSE.
C
      SINTI = 0.
      COSTI = 1.
      SIND  = 0.
      COSD  = 1.
      XCOR(1)=0.
      YCOR(1)=0.
      ZCOR(1)=0.
      XCOR(3)=0.
C  JG IS STARTING OFFSET OF GB(J)
C  I IS THE INDEX OF THE FIELD POINT
C  J IS THE INDEX OF THE INFLUENCING PANEL
C
C  INITIALIZE SKIP LOGIC CHECK ON VELOCITY CONTRIBUTION
C  OF SUBSEQUENT PANELS OF A SINGLE RING
      DO 20 I=1,NFLD
20    SVN(I) = 0.
C
C  SETUP LOCAL GEOMETRY FOR INFLUENCING PANEL OF RING JR
C
      DO 60 N=1,JR
      J = JG+N
      XBTJ = XPT(J)
      YBTJ = YPT(J)
      ZBTJ = ZPT(J)
      TAND=TAN(DELTA(J))
      COST=COS(THET(J))
      SINT=SIN(THET(J))
      GBJ = GB(J)
C
C  FOR INLET, SET TO GREATER OF BTINLT OR PANEL SLOPE
C
      BETAL = BETAO
      NOINLT = .NOT.INLTST(J,OPEN)
      IF (NOINLT) GO TO 30
      IF (BETAL.GT.BTINLT) BETAL=BTINLT
      IF (BTINLT*TAND.GT.1.) BETAL=0.99/TAND
C
30    NP1 = N+1
      LM1 = L-1
      XC1=XC(LM1)
      YC1=YC(LM1,NP1)
      ZC1=ZC(LM1,NP1)
C
C  CALCULATION OF PANEL CORNER POINTS IN PANEL COORDINATE SYSTEM
C              K= 1        LM1= L-1        NP1= N+1
C                 2             L               N+1
C                 3             L-1             N
C                 4             L               N
C
      XCOR(2)=XC(L)-XC1
      XCOR(4)=XCOR(2)
      DO 40 K=2,4
      LM1 = L
      NP1 = N+1
      IF (K.GE.3) NP1=N
      IF (K.EQ.3) LM1=L-1
      DELY=YC(LM1,NP1)-YC1
      DELZ=ZC(LM1,NP1)-ZC1
      YCOR(K)=DELY*COST+DELZ*SINT
      ZCOR(K)=DELZ*COST-DELY*SINT
40    CONTINUE
      CX=XCOR(2)
C             LOOP COLUMNWISE THROUGH ALL FIELD POINTS
      DO 50 I=1,NFLD
      LZERO = .FALSE.
      IF (LSKP(I).AND.NOINLT) GO TO 50
      XPTI = XFP(I)
      YPTI = YFP(I)
      ZPTI = ZFP(I)
      CALL PANVEL (UBI,VBI,WBI,ANI,IXZSYM)
      U(I) = U(I)+UBI*GBJ
      V(I) = V(I)+VBI*GBJ
      W(I) = W(I)+WBI*GBJ
      SVN(I) = SVN(I)+UBI*UBI+VBI*VBI+WBI*WBI
      IF (I.GT.1) LZERO=.TRUE.
50    CONTINUE
60    CONTINUE
C
C  CHECK WHETHER CONTRIBUTION OF RING IS ZERO.
      IF (SUBSON) GO TO 110
      DO 70 I=1,NFLD
```

Figure C-1(x)

```
70    LSKP(I) = SVM(I).EQ.0.
110   CONTINUE
      RETURN
      END


      SUBROUTINE FORMOM(XPT,YPT,ZPT,THET,DELTA,AREA,CP,XC)
C
C  CALCULATE THE FORCE AND MOMENT COEFFICIENTS ON A NON-CIRCULAR BODY
C
      DIMENSION XPT(1),YPT(1),ZPT(1),THET(1),DELTA(1),AREA(1),CP(1)
     *    ,XC(1)
      COMMON /BFORCX/ CLX(50),CMX(50),CNX(50),DXC(50)
      COMMON /BOPTNS/
     1    J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2    ,INLET,IPLOT(4),IPRT(5),IUVW,ASTART,XWLE
     3    ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,RODL,IZ1(12)
      COMMON /CFORCE/ CZX(80),CYX(80),XCP(81),DWDXCP(81),WCP(81)
     *    ,UT(81),VT(81),WT(81)
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *    ,CDC,XMOM,ESTRMX,DX,DFLX,VSTORE,ESTLOC
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *    ,NBODY,KFUS,KRADX(5),KFORX(5),IAC(5),IYC(5),IZC(5)
     *    ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *    ,IAR,IAN,IUR,IGR(7),IVR,IU,IV,IW,IVA,IWA,ICP,IPHI,ITR
     *    ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IA0,ID0,ISK0,IZZ(4),NRING,IPOW(50)
      COMMON /FORM  / CN,CT,CM,CYB,CLROLL,CNYAW,CL,CD,DXN
      COMMON /HEAD  / TITLE1(20),TITLE2(20)
      COMMON /IFORCE/ NDAMP,NEUSTR,NEMP,NGAM,NSEG ,NHSEGO,NROLL
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1    ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
C
      CN=0.
      CT=0.
      CM=0.
      CYB=0.
      CLROLL=0.
      CNYAW=0.
C
C  CALCULATE THE FORCES AND MOMENT ACTING ON EACH PANEL AND SUM OVER
C  THE ENTIRE COMPONENT
C  I IS PANEL NUMBER,  K IS RING NUMBER  AND  M IS SEGMENT NUMBER
C
      I = 0
      K = 0
      DO 170 M=1,KFUS
      NR = KFORX(M)-1
      JR = KRADX(M)-1
      DO 170 L=1,NR
      K = K+1
      KPM = K+M
      DX = XC(KPM)-XC(KPM-1)
C     XCP(K) = XC(KPM-1)+0.5*DX
      DXC(K) = DX
      SDCY = 0.0
      SDCZ = 0.0
      SDCL = 0.0
       DCM= 0.0
      SDCN = 0.0
      DO 160 J=1,JR
      I = I+1
      XP=XPT(I)
      YP=YPT(I)
      ZP=ZPT(I)
      SIND = SIN(DELTA(I))
      COSD = COS(DELTA(I))
      SINT = SIN(THET(I))
      COST = COS(THET(I))
      IF (J.EQ.1) XCP(K)=XP
      CNORM=-CP(I)*AREA(I)
C
C  FORCE AND MOMENT SIGN CONVENTIONS
```

```
C          CNORM  = POSITIVE OUTWARD NORMAL FORCE ON PANEL
C          CXB    = POSITIVE AFT
C          CYB    = POSITIVE TO RIGHT
C          CZB    = POSITIVE UP
C          CMB    = POSITIVE NOSE UP
C          CNYAW  = POSITIVE NOSE RIGHT VIEWING FORWARDS
C          CLROLL = POSITIVE CLOCKWISE VIEWING FORWARDS
C
      DCXB = -CNORM*SIND
      DCYB = -CNORM*COSD*SINT
      DCZB =  CNORM*COSD*COST
      DCMB    = -DCZB*(XP-REFX)+DCXB*(ZP-REFZ)
      DCNYAW = -DCYB*(XP-REFX)+DCXB*YP
      DCLROL = -DCZB*YP         +DCYB*(ZP-REFZ)
      THETP=ATAN2(ZP,YP)*57.29578
      IF (THETP.LT.0.0) THETP=THETP+360.0
      CN=CN+DCZB
      CT=CT+DCXB
      CM=CM+DCMB
      CYB = CYB+DCYB
      CNYAW = CNYAW+DCNYAW
      CLROLL = CLROLL+DCLRO
      SDCY = SDCY+DCYB
      SDCZ = SDCZ+DCZB
      SDCL = SDCL+DCLROL
      SDCM = SDCM+DCMB
      SDCN = SDCN+DCNYAW
160   CONTINUE
C
C                SAVE INTEGRATED RING NORMAL AND SIDE FORCE COEFFICIENTS
C
      CZX(K) = SDCZ/(DX*REFA)
      CYX(K) = SDCY/(DX*REFA)
      CLX(K) = SDCL/(REFA*REFD)
      CMX(K) = SDCM/(REFA*REFD)
      CNX(K) = SDCN/(REFA*REFD)
170   CONTINUE
      NSEG = K
C
C  STORE BODY AND WING FORCES AND MOMENT
C
C
C  COMPUTE NORMAL AND TANGENTIAL (AXIAL) FORCE, PITCHING MOMENT,
C  LIFT AND DRAG COEFFICIENTS, AND COMPONENT CENTER OF PRESSURE
C
C                CHECK FOR SYMMETRIC CONFIGURATION
      SYM = 2.0
      IF (IXZSYM.NE.0) SYM=1.0
      CN=SYM*CN/REFA
      CT=SYM*CT/REFA
      CM=SYM*CM/(REFA*REFD)
      CYB=CYB/REFA
      CNYAW=CNYAW/(REFA*REFD)
      CLROLL=CLROLL/(REFA*REFD)
      IF (IXZSYM.EQ.0) CYB=0.
      IF (IXZSYM.EQ.0) CLROLL=0.
      IF (IXZSYM.EQ.0) CNYAW=0.
C
      CL=CN*COSAC-CT*SINAC
      CD=CN*SINAC+CT*COSAC
      DXN=0.
      IF (CN.NE.0.) DXN=-CM/CN
      RETURN
      END



      SUBROUTINE FREFSH (NS,SX,SR)
C
C     THIS SUBROUTINE CALCULATES THE LOCATION AT WHICH THE CIRCULAR
C          STORE NOSE SHOCK WHICH IS REFLECTED BY THE FUSELAGE STRIKES
C          THE STORE AND THE VALUE OF BETA TO USE FOR THE IMAGE STORE
C          NOSE
```

Figure C-1(y)

```
C
C
      LOGICAL IMAGE
      DIMENSION SX(1),SR(1),IA(1)
      COMMON /BSWINT/ IP,XPTIP,YPTIP,ZPTIP,DELTIP,DRDXS,YBS,ZBS,BTNOSN
      COMMON /CONFIG/ NFU,NPY,NSTRS,NRACK
      COMMON /CONSTS/ PI,DTOR
      COMMON /OUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *     ,CDC,XMOM,FSTRMX,DX,DFLX,VSTORE,ESTLGC
      COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101),
     1      FRMAX,FLTHC,NFPOLY,FXEND(7),FCOEF(7,7)
      COMMON /MULSTR/ NFSHPT,NFUSHP,NEJGB,LASTEJ,LASTF,LASTA
     1  ,IDEJ,IDF,ID2
      COMMON/STORIC/BTNOSC,RSSC
      COMMON/STORIF/ IMFSTR,BTNOSF,RSS,YBN,ZBN,SFAC
      COMMON /STORIM/ IMSTOR,BTNOSE,XWN,YWN,ZWN,IMAGE
      COMMON/WGEOM/XBWOC,ZBWO,CRW,ZDIHED
      COMMON          A(12000)
      EQUIVALENCE (IA(1),A(1))
C
C     CALCULATE RADIAL DISTANCE FROM STORE NOSE TO FUSELAGE AXIS
C
      XBN=XWN+XBWOC
      YBN=YWN
      ZBN=ZWN+ZBWO
      RBN=SQRT(YBN*YBN+ZBN*ZBN)
      XZ=-XBN
      IF (NFU.EQ.2) GO TO 100
C
C     CALCULATE CIRCULAR STORE SHOCK LOCATION AT CIRCULAR FUSELAGE
C             SURFACE, SHOCK ANGLE, AND FUSELAGE SURFACE SLOPE ---------
C
      CALL SWINT(XZ,RBN,NS,SX,SR,FXEND,FCOEF,NFPOLY,FLTHC,FRMAX,DRDXS,
     1DRDXB,XBS,RBS,IMFSTR)
      IF (IMFSTR.EQ.0) RETURN
      XBS=-XBS
C
C     CALCULATE SHOCK WAVE REFLECTION ANGLE
C
      THETAR=(ATAN(DRDXS)+2.0*ATAN(DRDXB))/DTOR
      GO TO 120
C
C     CALCULATE CIRCULAR STORE SHOCK LOCATION AT NONCIRCULAR FUSELAGE
C             SURFACE, SHOCK ANGLE AND SURFACE SLOPE ------------------
C
100   IDO = IA(IDF+74)
      IAO  = IA(IDF+73)
      IXPT = IA(IDO+42)+IAO
      IYPT = IA(IDO+43)+IAO
      IZPT = IA(IDO+44)+IAO
      ITH  = IA(IDO+45)+IAO
      IDEL = IA(IDO+46)+IAO
      IXC  = IA(IDO+23)+IAO
      IKR  = IDO+13
      IKX  = IDO+18
      NFUS = IA(IDO+12)
      BLN = A(IDO+180)
C
C  COMPUTE XBS AND RBS OF SHOCK INTERSECTION WITH FUSELAGE
C
      CALL SWINTE (XZ,YBN,ZBN,NS,SX,SR,BLN,XBS,RBS,NFUS,IMFSTR
     *     ,IA(IKX),IA(IKR),A(IXPT),A(IYPT),A(IZPT),A(ITH),A(IDEL),A(IXC))
      IF (IMFSTR.EQ.0) RETURN
      XBS = -XBS
C
C  COMPUTE REFLECTION ANGLE (ADD TANGENTS)
C
      TAN2D = TAN(2.*DELTIP)
      DRDXR = -(TAN2D-DRDXS)/(1.+TAN2D*DRDXS)
      THETAR = ATAN(DRDXR)
C
C     LOCATE STORE C.G. IN BODY SYSTEM AND CALCULATE RADIAL DISTANCE
C           TO SURFACE ASSUMING C.G. LIES IN PLANE CONTAINING SHOCK AND
C           FUSELAGE AXIS
C
120   CONTINUE
      XBCG=VAR(7)
      YBCG=VAR(8)
      ZBCG=VAR(9)
      RCG=SQRT(YBCG*YBCG+ZBCG*ZBCG)
C
C     DETERMINE INTERSECTION OF REFLECTED SHOCK WAVE AND LINE FROM STORE
C           NOSE THROUGH ABOVE POINT
C
      TALF=(RCG-RBN)/(XBN-XBCG)
      ALF=ATAN(TALF)/DTOR
      IF (THETAR.GT.ALF) GO TO 150
140   IMFSTR=0
      RETURN
150   IF (THETAR.LE.89.5.OR.THETAR.GE.90.5) GO TO 160
      XBR=XBS
      GO TO 170
160   DRDXR=TAN(THETAR*DTOR)
      XBR=(RBN-RBS+XBN*TALF-XBS*DRDXR)/(TALF-DRDXR)
170   RBR=RBN+TALF*(XBN-XBR)
C
C     CALCULATE DISTANCE MEASURED ALONG AXIS TO REFLECTED SHOCK AND
C           SEE IF THIS POINT IS BEHIND BASE OF STORE
C
      ALFR=ALF*DTOR
      XSR=(XBN-XBR)*COS(ALFR)+(RBR-RBN)*SIN(ALFR)
      IF (XSR.GE.FSTLGC) GO TO 140
      IMFSTR=1
C
C     CALCULATE VALUE OF BETA TO USE FOR IMAGE STORE NOSE
C
      RSSC=RBN
      RSS=RSSC-RBS*RBS/RSSC
      BTNOSC=XSR/RSSC
      BTNOSF=XSR/RSS
      SFAC=RBS*RBS/(RSSC*RSSC)
      RETURN
      END




      SUBROUTINE FRSTRT (IO,KODE,LASTA)
C
C  ROUTINE TO SAVE OR REST RE REQUIRED PROGRAM INFORMATION FOR SOURCE
C  PANELING METHOD TO RESTART CONFIGURATION ANALYSIS.
C        IO    = FILE UNIT NUMBER FOR DATA STORAGE.
C        KODE  = 1, SAVE CONTROL INTEGERS AND VARIABLES
C        KODE  = 2, RESTORE CONTROL INTEGERS AND VARIABLES
C        KODE  = 3, SAVE ABOVE AND AIC,U,V,W MATRICES
C        KODE  = 4, RESTORE ABOVE AND AIC,U,V,W MATRICES
C        KODE  = 5, RESTORE CONTROL INTEGERS AND VARIABLES INTO
C                   CORE AFTER EXISTING DATA IN BLANK COMMON
C        KODE  = 6, RESTORE ABOVE AND SKIP PAST TAPE8 + TAPE9 DATA
C  MULTIPLE CASES MAY BE STACKED BACK TO BACK.  THE CALLING PROGRAM IS
C  RESPONSIBLE FOR PROPER POSITIONING OF SEQUENTIAL CASES.
C
      COMMON /RGEOM / GEOMB(339)
      COMMON /RINLET/ AINLET(38)
      COMMON /RINSHK/ AINSHK(189)
      COMMON /ROPTNS/ IOPTS(50)
      COMMON /RSHOCK/ SHOCKB(240)
      COMMON /DIMENS/ IDIM(79),NRING,IROW(50)
      COMMON /HEAD  / TITLE(40)
      COMMON /PARAM / RPARAM(12)
      COMMON          A(12000)
      DIMENSION IA(130)
      EQUIVALENCE (NTAP7,IDIM(47)),  (INLET,IOPTS(18))
      EQUIVALENCE (IA(1),A(1),LA(1)),  (IUVW,IOPTS(28))
      LOGICAL INLET,SKIP,LA(1),INLETA
C
      GO TO (100,200,100,200,300,300),KODE
C
```

Figure C-1(z)

```
C  SAVE COMMON BLOCK AND TAPE7 ON TAPE-10
100   CONTINUE
      CALL IOWRIT (10,IDIM,130)
      CALL IOWRIT (10,BPARAM,12)
      CALL IOWRIT (10,IOPTS,50)
      CALL IOWRIT (10,GEOMB,339)
      CALL IOWRIT (10,TITLE,40)
      CALL IOWRIT (10,SHOCKB,240)
      IF (INLET) CALL IOWRIT (10,AINLET,38)
      IF (INLET) CALL IOWRIT (10,AINSHK,189)
      CALL IOWRIT (10,A,NTAP7)
      IF (KODE.NE.3)                      RETURN
C
C  COPY AN AND U,V,W  MATRICES FROM TAPE9 AND TAPE8
      REWIND 8
      REWIND 9
      CALL IOREAD (9,IROW,NRING)
      SKIP = IUVW.NE.0
      DO 150 JB=1,NRING
      JR = IROW(JB)
      DO 150 IB=JB,NRING
      IR = IROW(IB)
      NA = IR*JR
C              AN - MATRIX
      CALL IOREAD ( 9,A,NA)
      CALL IOWRIT (10,A,NA)
C           U,V,W
      IF (SKIP) GO TO 150
      CALL IOREAD ( 8,A,3*NA)
      CALL IOWRIT (10,A,3*NA)
150   CONTINUE
      RETURN
C
C  RESTORE DATA FROM TAPE - 10  ----------------------------------------
C
200   CONTINUE
      CALL IOREAD (10,IDIM,130)
      CALL IOREAD (10,BPARAM,12)
      CALL IOREAD (10,IOPTS,50)
      CALL IOREAD (10,GEOMB,339)
      CALL IOREAD (10,TITLE,40)
      CALL IOREAD (10,SHOCKB,240)
      IF (INLET) CALL IOREAD (10,AINLET,38)
      IF (INLET) CALL IOREAD (10,AINSHK,189)
      CALL IOREAD (10,A,NTAP7)
      REWIND 7
      CALL IOWRIT ( 7,A,NTAP7)
      LASTA = NTAP7
      IF (KODE.NE.4)                      RETURN
C
C  COPY AN AND U,V,W  MATRICES ONTO TAPE9 AND TAPE8
C
      REWIND 8
      REWIND 9
      CALL IOWRIT (9,IROW,NRING)
      SKIP = IUVW.NE.0
      DO 250 JB=1,NRING
      JR = IROW(JB)
      DO 250 IB=JB,NRING
      IR = IROW(IB)
      NA = IR*JR
C              AN - MATRIX
      CALL IOREAD (10,A(LASTA+1),NA)
      CALL IOWRIT ( 9,A(LASTA+1),NA)
C           U,V,W
      IF (SKIP) GO TO 250
      CALL IOREAD (10,A(LASTA+1),3*NA)
      CALL IOWRIT ( 8,A(LASTA+1),3*NA)
250   CONTINUE
      RETURN
C
C  RESTORE DATA FROM TAPE-10 TO BLANK COMMON ---------------------------
C
300   CONTINUE
      IDO = LASTA
C                          IDIM
      CALL IOREAD (10,IA(IDO+1),130)
      IA(IDO+74) = IDO
C                          NRING
      IF (KODE.EQ.6) NR=IA(IDO+80)
      IPO = IDO+130
C                          BPARAM
      CALL IOREAD (10,A(IPO+1),12)
      ISKO = IPO+12
C                          IOPTS
      CALL IOREAD (10,A(ISKO+1),50)
      INLETA = LA(ISKO+1A)
      SKIP = IA(ISKO+28).NE.0
      ISKO = ISKO+50
C                          GEOMB
      CALL IOREAD (10,A(ISKO+1),339
      ISKO = ISKO+339
C                          TITLE
      CALL IOREAD (10,A(ISKO+1),40)
      ISKO = ISKO+40
C                          SHOCKB
      CALL IOREAD (10,A(ISKO+1),240)
      IA(IDO+75) = ISKO
      IAO = ISKO+240
      IA(IDO+73) = IA
C                          INLET
      IF (INLETA) CALL IOREAD (10,AINLET,38
      IF (INLETA) CALL IOREAD (10,AINSHK,189)
C                          A        NTAP7
      CALL IOREAD (10,A(IAO+1),IA(IDO+47))
      LASTA = LASTA+IAO+IA(IDO+47
C
C  FOR KODE=6, READ PAST U,V,W INFLUENCE MATRICES.
      IF (KODE.EQ.5)                      RETURN
      DO 350 JB=1,NR
      JR = IDO+JB+80
      DO 350 IB=JB,NR
      IR = IDO+IB+80
      NA = IA(JR)*IA(IR)
      CALL IOREAD (10,A(LASTA+1),NA
      IF (SKIP) GO TO 350
      CALL IOREAD (10,A(LASTA+1),3*NA
350   CONTINUE
      RETURN
      END



      SUBROUTINE FXBOD(DCYX,DCZX,FBOD,IBOD,NBOD)
C
C  ROUTINE TO SUM AXIAL FORCE DISTRIBUTIONS OF RINGS BETWEEN STORE
C  BODY AND STORE FIN SECTIONS.
C
      DIMENSION DCYX(1),DCZX(1),IBOD(5),FBOD(5,5)
      COMMON /BFORCX/ CLX(50),CMX(50),CNX(50),DXC(50)
C
      I2 = 0
      DO 120 J=1,NBOD
      I1 = I2+1
      I2 = IBOD(J)-1
      SCY = 0.
      SCZ = 0.
      SCL = 0.
      SCM = 0.
      SCN = 0.
C
C  SUM FORCES AND MOMENTS IN SECTION OF BODY
C
      DO 110 I=I1,I2
      SCY = SCY+DCYX(I)*DXC(I)
      SCZ = SCZ+DCZX(I)*DXC(I)
      SCL = SCL+CLX(I)
```

Figure C-1(aa)

AD-A099 332 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 20/4
PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLU--ETC(U)
NOV 80 J MULLEN, F K GOODWIN, M F DILLENIUS F33615-76-C-3077
UNCLASSIFIED NEAR-TR-210-VOL-4 AFWAL-TR-80-3032-VOL-4 NL

3 OF 4
AD
A099332

```
      SCM = SCM+CMX(I)
110   SCN = SCN+CNX(I)
C  DEFINE SECTION FORCES AND MOMENTS IN  ARRAY - FBOD
      FBOD(1,J) = SCY
      FBOD(2,J) = SCZ
      FBOD(3,J) = SCL
      FBOD(4,J) = SCM
      FBOD(5,J) = SCN
120   CONTINUE
      RETURN
      END


      SUBROUTINE IMAGEV (GB,VB,IMAGE,JG,JR)
C
C  ROUTINE TO COMPUTE INFLUENCE OF JB*TH RING OF IMAGE STORE LOCATED AT
C  (VAR7,VAR8,VAR9) ON REAL STORE.
C
      DIMENSION GB(1),VB(1)
      COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4)
     1   ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,JP
      COMMON /BPHII / ZIP,PHII,CPHI,SPHI,C2PHII,S2PHII
      COMMON /BSHOCK/ SHOCKB(240)
      COMMON /BSWINT/ IP,XPTIP,YPTIP,ZPTIP,DELTIP,ORDXS,YBS,ZBS,BTNOSN
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *   ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAV,NAS,NASMK,NAFLD,IAO,IDO,ISKD,IYIM,IZIM,ISVN,ISKP
     *   ,NRING,IROW(50)
      COMMON           A(12000)
      LOGICAL IMAGE,LA(1)
      EQUIVALENCE (LA(1),A(1)),(SHOCKB(37),XSMLDR)
C
C  DETERMINE SEGMENT (IFU) AND RING INDEX (L)
C            (FOR STORE, ONLY ONE SEGMENT ALLOWED HERE)
C
      JGP = 0
      IFU = 1
      KFUSOR = KFORX(IFU)
      DO 100 L=2,KFUSOR
      IF (JG.EQ.JGP) GO TO 110
100   JGP = JGP+JR
      L = KFUSOR
110   IXCO = IXC(IFU)
      IYCO = IYC(IFU)
      IZCO = IZC(IFU)
C
C  STARTING AND ENDING FIELD POINTS
C
      IXF = IXPT+JG
      IYF = IYIM+JG
      IZF = IZIM+JG
      IUF = IU+JG
      IVF = IV+JG
      IWF = IW+JG
      NFLD = NBODY-JG
C
C  INITIALIZE INCREMENTAL VELOCITIES
C
      JU = IU-1
      NB3 = 3*NBODY
      DO 120 I=1,NB3
      JU = JU+1
120   A(JU) = 0.0
C
C  COMPUTE VELOCITIES OF JB(TH) RING ON ALL FIELD POINTS
C
      BETAO = BTNOSN
      CALL FLDVL2 (A(IXPT),A(IYPT),A(IZPT),A(ITH),A(IDEL),GB,A(IXF)
     *   ,A(IYF),A(IZF),A(IUF),A(IVF),A(IWF),A(ISVN),LA(ISKP),NFLD
     *   ,A(IXCO),A(IYCO),A(IZCO),KFUSOR,IXZSYM,JR,JG,L)
C
C  INFLUENCE OF IMAGE STORE ON REAL STORE IS COMPUTED FROM
C            (V,W)IMAGE = (V,-W)REAL
C
      IMAGE = .FALSE.
      DO 130 I=1,NFLD
      LJG = ISKP+I-1
      IF (LA(LJG)) GO TO 130
      IJG = I+JG
      JV = IV+IJG-1
      JW = IW+IJG-1
C
C  IMAGE TRANSFORMATION INCLUDING ROTATIONS FORWARD AND BACK
C
      V = A(JV)
      W = A(JW)
      A(JV) = C2PHII*V-S2PHII*W
      A(JW) = -S2PHII*V-C2PHII*W
C
C  ADD NORMAL VELOCITIES TO BODY B.C. - VB
C
      VNORMI = VNORM(IJG,A(IU),A(IV),A(IW),A(ITH),A(IDEL))
      VB(IJG) = VB(IJG)-VNORMI
      IMAGE = .TRUE.
130   CONTINUE
      RETURN
      END


      SUBROUTINE IMAGFN(XCPT,YCPT,ZCPT,UIFIN,VIFIN,WIFIN,XWLE)
C
C  ROUTINE TO COMPUTE INFLUENCE OF IMAGE STORE BODY ON FIN CONTROL
C  POINTS.  VELOCITIES ARE ACTUALLY COMPUTED FOR REAL STORE AT IMAGE
C  STORE LOCATIONS AND TRANSFORMED BACK ONTO REAL STORE FINS.
C
      LOGICAL IMAGE
      COMMON /BPHII / ZIP,PHII,CPHI,SPHI,C2PHII,S2PHII
      COMMON /BSWINT/ IP,XPTIP,YPTIP,ZPTIP,DELTIP,ORDXS,YBS,ZBS,BTNOSN
      COMMON /DIMENS/ ID(130)
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1   ,IDEJ,IDF,ID2
      COMMON /STORIM/ IMSTOR,BTNOSE,XWN,YWN,ZWN,IMAGE
      COMMON           A(12000)
C
      UIFIN = 0.0
      VIFIN = 0.0
      WIFIN = 0.0
      IF (.NOT.IMAGE) GO TO 999
C
C  LOCATE FIN CONTROL POINT ON IMAGE STORE RELATIVE TO ELLIPTIC
C  SOURCE PANELS COORDINATES
C
      XPT = XWLE+XCPT
      YI = CPHI*YCPT+SPHI*ZCPT
      ZI = SPHI*YCPT-CPHI*ZCPT+ZIP
C
      YIM = CPHI*YI-SPHI*ZI
      ZIM = SPHI*YI+CPHI*ZI
C
C  TEST IF PANEL INFLUENCED BY IMAGE STORE
C
      BTCPT = SQRT(YIM*YIM+ZIM*ZIM)/XPT
      IF (BTCPT.LT.BTNOSN) GO TO 999
C
C  COMPUTE IMAGE STORE VELOCITY INFLUENCE ON FINS
C
      CALL FLDVEL (XPT,YIM,ZIM,SVN,LSKP,UP,VP,WP,1,ID,NEJGB)
C
C  TRANSFORM VELOCITIES TO REFLECTION AXIS, SET W=-W, AND
C  TRANSFORM BACK TO STORE COODRDINATES
C
      UIFIN = UP
```

Figure C-1(bb)

```
      VIFIN = C2PHII*VP-S2PHII*WP
      WIFIN = -S2PHII*VP-C2PHII*WP
  999 RETURN
      END


      SUBROUTINE IMAGYZ(YPT,ZPT,YIM,ZIM,NBODY,VAR12,YIMO,ZIMO,YPTO,ZPTO)
C
C  ROUTINE TO COMPUTE CONTROL POINT LOCATIONS OF IMAGE STORE AS SEEN
C  FROM THE BODY FIXED REAL STORE.
C
C  YIMO,ZIMO,YPTO,ZPTO ARE IN FUSELAGE INERTIAL COORDINATES
C  YIM,ZIM,YPT,ZPT     ARE IN SOURCE PANEL COORDINATES OF STORE
C
      DIMENSION YPT(1),ZPT(1),YIM(1),ZIM(1)
      COMMON /BPHII / ZIP,PHII,CPHI,SPHI,C2PHII,S2PHII
      PHII = ATAN2(YIMO-YPTO,-(ZIMO-ZPTO))
      ZIP = SIN(PHII)*(YIMO-YPTO)-COS(PHII)*(ZIMO-ZPTO)
      PHI = PHII-VAR12
      CPHI = COS(PHI)
      SPHI = SIN(PHI)
      C2PHII = COS(2.*PHI)
      S2PHII = SIN(2.*PHI)
      DO 100 I=1,NBODY
C  LOCATE IMAGE OF YPT,ZPT TO IMAGE\STORE AXIS
      YI =  CPHI*YPT(I)-SPHI*ZPT(I)
      ZI = -SPHI*YPT(I)-CPHI*ZPT(I)+ZIP
C  IMAGE POINT IN LOCAL STORE COORDINATE SYSTEM
      YIM(I) =  CPHI*YI+SPHI*ZI
      ZIM(I) = -SPHI*YI+CPHI*ZI
  100 CONTINUE
      RETURN
      END


      SUBROUTINE INSVEL (NN,XFP,RFP,CTHET,STHET,BETAAL,BETASL,USD,VSD,
     1 WSD,FAC)
C
C     THIS SUBROUTINE CALCULATES THE IMAGE STORE VELOCITIES
C
C
      COMMON /SORDUB/ BETAA,SRS(150),VRATSA,BETAS,BSS,DSW(150),
     1 DSV(150),XS(150),XD(150),BETAAI,BETASI,BSSI
      COMMON /TRMSSD/ ARG,ROOT,ACOSH
C
C     SOURCE VELOCITIES
C
      SAX=0.0
      SRAD=0.0
      DO 32 NN=1,NM
      BARR=RFP*BETAAL
      ARG=(XFP-XS(NN))/BARR
      IF (ARG.LE.1.0) GO TO 33
      CALL SDTRMS
      SS=SRS(NN)
      SAX=SAX-SS*ACOSH
   32 SRAD=SRAD+SS*BETAAL*ROOT
   33 CONTINUE
      SAX=SAX/VRATSA
      SRAD=SRAD/VRATSA
C
C     DOUBLET VELOCITIES
C
      DWAX=0.0
      DVAX=0.0
      DWRAD=0.0
      DVRAD=0.0
      DWTAN=0.0
      DVTAN=0.0
      IF (FAC.LE.0.0) GO TO 41
      DO 40 NN=1,NM
      BSRR=RFP*BETASL
      ARG=(XFP-XD(NN))/BSRR
      IF (ARG.LE.1.0) GO TO 41
      DW=DSW(NN)
      DV=DSV(NN)
      CALL SDTRMS
      TA=0.5*BETASL*BETASL
      TB=BETASL*ROOT
      DWAX=DWAX+DW*TB*CTHET
      DVAX=DVAX+DV*TB*STHET
      TC=ARG*ROOT
      TD=TA*(TC-ACOSH)
      TE=TA*(TC+ACOSH)
      DWRAD=DWRAD-DW*TE*CTHET
      DVRAD=DVRAD-DV*TE*STHET
      DWTAN=DWTAN-DW*TD*STHET
   40 DVTAN=DVTAN+DV*TD*CTHET
   41 CONTINUE
C
C     SUM UP IMAGE VELOCITIES
C
      USD=SAX+FAC*(DWAX+DVAX)
      RADSD=SRAD+FAC*(DWRAD+DVRAD)
      TANSD=FAC*(DWTAN+DVTAN)
      VSD=-RADSD*STHET+TANSD*CTHET
      WSD= -RADSD*CTHET-TANSD*STHET
      RETURN
      END


      SUBROUTINE INLBET (BETAI,XS,RS,Y,Z)
C
C  ROUTINE TO INTERPOLATE IN A WEDGE SHAPED INLET SHOCK
C  ASSUMPTIONS
C    1 - BELOW INLET, TRAVERSES ARE COMPUTED AT CONSTANT Y-VALUE (PHI=0)
C    2 - ABOVE INLET (PHI=180), USE BETA(INFINITY) BETWEEN +-YINLT
C    3 - LINEAR R VARIATION WITH THTS
C    4 - SYMMETRIC ABOUT CENTERLINE
C
      COMMON /RINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI
     *  ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BINSHK/ NIS,XISHLD,EALPI,THTSI,MAXSHI,NINL(8),PHINL(8)
     *  ,YINL(8),XINL(80),RINL(80)
      COMMON /CONSTS/ PI,DTOR
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1   ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      DATA ALF/0.0/
C
C  CHECK WHETHER POINT IS IN WEDGE OR IN CONE REGION
C
      YI = ABS(Y)
      ZI = Z-ZINLT
      IF (YI.GT.YINLT) GO TO 130
      RS = ABS(ZI)
      IF (Z.GT.ZINLT) GO TO 120
C
C  BELOW INLET.  INTERPOLATE IN VERTICAL TRAVERSES ----------------------
C
      IF (NIS.GT.2) GO TO 100
      CALL XVSR (RINL,XINL,NINL,RS,XS,ALF)
C             IF INBOARD OF INLET INSIDE CORNER, YCPI, SET RS TO
C             DISTANCE FROM INSIDE LEADING EDGE
      IF (YI.LT.YCPI) RS=SQRT(ZI*ZI+(YI-YCPI)**2)
      BETAI = XS/RS
      GO TO 200
C
C  USE TWO POINT INTERPOLATION BETWEEN Y-VALUES
C             FIND CLOSEST  Y TRAVERSES BRACKING POINT
C
  100 IS = (NIS+1)/2
      I2 = 1
```

Figure C-1(cc)

```
      DO 105 J=2,IS
      I1 = I2
      I2 = MAXSHI*(J-1)+1
      IF (YI.LT.YINL(J)) GO TO 110
105   CONTINUE
      J = IS
110   CALL XVSR (RINL(I1),XINL(I1),NINL(J-1),RS,XSJM1,ALF)
      CALL XVSR (RINL(I2),XINL(I2),NINL(J),RS,XSJ,ALF)
      XS = XSJ+(YI-YINL(J))*(XSJM1-XSJ)/(YINL(J-1)-YINL(J))
      IF (YI.LT.YCPI) RS=SQRT(ZI*ZI+(YI-YCPI)**2)
      BETAI = XS/RS
      GO TO 200
C
C  ABOVE INLET IN WEDGE - BETA(INFINITY)
C
120   BETAI = BETA
      XS = BETA*RS
      GO TO 200
C
C  USE CONE PORTION OF INLET SHOCK -----------------------------------
C  INTERPOLATE TO FIND POLAR ANGLE OF FIELD POINT
C
130   YI = YI-YINLT
      ZI = Z-ZINLT
      RS = SQRT(YI*YI+ZI*ZI)
      THTS = ATAN2(YI,-ZI)/DTOR
      IS = (NIS+1)/2
      I2 = 1
      IF (IS.GT.1) I2=MAXSHI*(IS-2)+1
      DO 140 J=IS,NIS
      I1 = I2
      I2 = MAXSHI*(J-1)+1
      IF (THTS.LT.PHINL(J)) GO TO 150
140   CONTINUE
      J = NIS
150   CONTINUE
      CALL XVSR (RINL(I1),XINL(I1),NINL(J-1),RS,XSJM1,ALF)
      CALL XVSR (RINL(I2),XINL(I2),NINL(J),RS,XSJ,ALF)
      XS = XSJ+(THTS-PHINL(J))*(XSJM1-XSJ)/(PHINL(J-1)-PHINL(J))
      BETAI = XS/RS
200   RETURN
      END


      LOGICAL FUNCTION INLTST(I,OPEN)
C
C  FUNCTION TO COMPARE PANEL NUMBER, I,  WITH TABLE OF INLET PANELS.
C  VALUE RETURNED FOR FUNCTION IS ,
C       .TRUE.,  IF I IS INLET PANEL
C       .FALSE., IF I IS NOT INLET PANEL OR INLET NOT EXIST
C  VALUE OF OPEN DETERMINES WHETHER PANEL IS FULLY OPEN (TRUE)
C  OR FULLY BLOCKED TO FLOW (FALSE).
C
      LOGICAL INLET,OPEN
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,RTINLT,YCPI
     *   ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BOPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BOOL,IZ1(12)
      IF (INLET) GO TO 100
C           NO INLET EXISTS
      INLTST = .FALSE.
      OPEN = .FALSE.
      RETURN
C           COMPARE PANEL NUMBERS
100   DO 110 J=1,NTINL
      IF (I.EQ.JINLT(J)) GO TO 120
110   CONTINUE
      INLTST = .FALSE.
      OPEN = .FALSE.
      RETURN
120   INLTST = .TRUE.
      OPEN = J.LE.NINLET
      RETURN
      END


      SUBROUTINE INTOST (XI,ETA,ZETA,X,Y,Z,DC)
C     SUBROUTINE TO TRANSFORM FROM INERTIAL TO STORE SYSTEM
C
      DIMENSION DC(3,3)
      X=XI*DC(1,1)+ETA*DC(2,1)+ZETA*DC(3,1)
      Y=XI*DC(1,2)+ETA*DC(2,2)+ZETA*DC(3,2)
      Z=XI*DC(1,3)+ETA*DC(2,3)+ZETA*DC(3,3)
      RETURN
      END


      SUBROUTINE INVER2(A,NSYS,N,NMAX,MMAX)
C     SUBROUTINE TO SOLVE SIMULTANEOUS EQUATIONS
C
      DIMENSION A(NMAX,MMAX),X(   6)
    5 SIGN=1.0
      NPI=N+1
      NMI=N-1
      NPLSY=N+NSYS
    9 DO 14 I=1,NMI
      IPI=I+1
      MAX=I
      AMAX=ABS(A(I,I))
      DO 10 K=IPI,N
      AKMAX=ABS(A(K,I))
      IF(AKMAX.LE.AMAX) GO TO 10
      MAX=K
      AMAX=AKMAX
   10 CONTINUE
      IF(AMAX.LT.1.0E-12) GO TO 16
      IF(MAX.EQ.I) GO TO 12
      DO 11 L=I,NPLSY
      TEMP=A(I,L)
      A(I,L)=A(MAX,L)
   11 A(MAX,L)=TEMP
      SIGN=-SIGN
   12 DO 14 J=IPI,N
      IF (A(J,I)) 30,14,30
   30 CONST=-A(J,I)/A(I,I)
      DO 13 L=I,NPLSY
   13 A(J,L)=A(J,L)+A(I,L)*CONST
   14 CONTINUE
      DO 15 I=1,N
      IF (A(I,I)) 15,16,15
   15 CONTINUE
      GO TO 18
   16 WRITE(6,100)
  100 FORMAT(///5X,34H***** MATRIX IS SINGULAR,   INVER2)
      STOP
   18 DO 21 I=NPI,NPLSY
      DO 20 KK=1,N
      K=NPI-KK
      X(K)=A(K,I)
      IF(K.EQ.N) GO TO 20
      J=K
   19 J=J+1
      X(K)=X(K)-A(K,J)*X(J)
      IF(J.NE.N) GO TO 19
   20 X(K)=X(K)/A(K,K)
      DO 21 J=1,N
   21 A(J,I)=X(J)
      RETURN
      END
```

Figure C-1(dd)

```
      SUBROUTINE IOREAD (IO,A,NA)
C
C  PERFORM AN UNFORMATTED READ FROM FILE  IO  OF ARRAY  A.
C     ******* CDC VERSION FOR BUFFERED I/O ********************************
C             USE OF BUFFERED IO STATEMENTS HERE ALSO ALLOWS THE
C             USER TO SET THE FILE BUFFER LENGTH TO ZERO IN THE
C             PROGRAM CARD (I.E.   TAPE8=0)
      DIMENSION A(NA)
C
C66   FORMAT (//,27H  IOREAD PARITY ERROR - IO=,I5,5H  NA=,I5
C    *   ,9H  UNITIO=,I5)
C67   FORMAT (//,28H  IOREAD END-OF-FILE - IO=,I5,5H  NA=,I5
C    *   ,9H  UNITIO=,I5)
C     IF (     IO.EQ.12) READ (IO) A
C     IF (.NOT.IO.EQ.12) BUFFER IN (IO,0) (A(1),A(NA))
C     UNITIO = UNIT(IO)
C     IF (UNITIO) 999,967,966
C966  WRITE (6,66) IO,NA,UNITIO
C     STOP 66
C967  WRITE (6,67) IO,NA,UNITIO
C     STOP 67
C999  RETURN
C
C     ******* ANSI FORTRAN VERSION ****************************************
C             TO USE, REMOVE  C  IN COLUMN ONE AND REPLACE ABOVE CARDS
      READ (IO) A
      RETURN
C
      END


      SUBROUTINE IOWRIT(IO,A,NA)
C
C  PERFORM AN UNFORMATTED WRITE TO FILE  IO  OF ARRAY A.
C     ******* CDC VERSION FOR BUFFERED I/O ********************************
C             USE OF BUFFERED IO STATEMENTS HERE ALSO ALLOWS THE
C             USER TO SET THE FILE BUFFER LENGTH TO ZERO IN THE
C             PROGRAM CARD (I.E.   TAPE8=0)
      DIMENSION A(NA)
C
C66   FORMAT(14H  IOWRIT ERROR,2I5,G12.4)
C
C     IF (     IO.EQ.12) WRITE (IO) A
C     IF (.NOT.IO.EQ.12) BUFFER OUT (IO,0) (A(1),A(NA))
C     UNITIO = UNIT(IO)
C     IF (UNITIO.GE.0.) GO TO 999
C     RETURN
C999  WRITE (6,66) IO,NA,UNITIO
C     STOP 66
C
C     ******* ANSI FORTRAN VERSION ****************************************
C             TO USE, REMOVE  C  IN COLUMN ONE AND REPLACE ABOVE CARDS
C
      WRITE(IO) A
      RETURN
      END


      SUBROUTINE IXBOD (XBOD,IBOD,NBOD,LFIN,XWLE1,XWLE2,RA1,RA2,RB1,RB2)
C
C  ROUTINE TO DEFINE LEADING AND TRAILING EDGES OF BODY-FIN SECTIONS
C  NOTES  ROUTINE RESTRICTED TO SINGLE SEGMENT BODY DESCRIPTION
C         DUE TO LIMITATIONS ON KFUSOR(1) SEARCH
C
      DIMENSION XBOD(5),IBOD(5)
      LOGICAL LFIN(5)
      COMMON /BGEOM / XFUS(51),ZFUS(51),FUSARD(51),FUSBY(51),FUSAZ(51)
     *  ,XJ(51),PHIK(33)
      COMMON /BOPTNS/ IOPTS(50)
      COMMON /DIMENS/ IDIM(130)
      COMMON            A(12000)
      EQUIVALENCE (IDIM(18),KFUSOR), (IDIM(23),IXC)
      EQUIVALENCE (IOPTS(10),NFUSOR)
      IFIN = 1
      KFORM1 = KFUSOR-1
      DO 130 I=1,NBOD
      DO 100 J=2,KFORM1
      IXP1 = IXC+J
      IX   = IXP1-1
      IXM1 = IX-1
      XM = 0.5*(A(IX)+A(IXM1))
      XP = 0.5*(A(IXP1)+A(IX))
      IF (XBOD(I).GE.XM.AND.XBOD(I).LT.XP) GO TO 120
  100 CONTINUE
      J = KFUSOR
  120 IBOD(I) = J
      IF (I.GT.1) IXLE=IBOD(I-1)+IXC-1
      IF (LFIN(I).AND.IFIN.EQ.1) XWLE1=A(IXLE)
      IF (LFIN(I).AND.IFIN.EQ.2) XWLE2=A(IXLE)
      IF (LFIN(I)) IFIN=IFIN+1
  130 CONTINUE
C
C  FIND MAXIMUM  A AND B  WITHIN SEGMENT FOR INTERFERENCE SHELL---------
C
      J1 = 1
      IFIN = 1
      DO 220 I=1,NBOD
      J2 = IBOD(I)
      IF (LFIN(I) .AND. IFIN.EQ.1) GO TO 140
      IF (LFIN(I) .AND. IFIN.EQ.2) GO TO 180
      GO TO 220
C
C  1ST FIN SEMI-AXES
C
  140 RA1 = 0.0
      RB1 = 0.0
      DO 170 J=J1,J2
C             INTERPOLATE FOR RA AND RB VERSUS XJ(J)
      DO 150 K=2,NFUSOR
      IF (XJ(J).LE.XFUS(K)) GO TO 160
  150 CONTINUE
      K = NFUSOR
C             INTERPOLATE BETWEEN VALUES
  160 DX = XFUS(K)-XFUS(K-1)
      IF (DX.NE.0.0) DX=(XJ(J)-XFUS(K-1))/DX
      RA1 = AMAX1(RA1,(FUSAZ(K)-FUSAZ(K-1))*DX+FUSAZ(K-1))
      RB1 = AMAX1(RB1,(FUSBY(K)-FUSBY(K-1))*DX+FUSBY(K-1))
  170 CONTINUE
      IFIN = 2
      GO TO 220
C
C  2ND FIN SEMI-AXES
  180 RA2 = 0.0
      RB2 = 0.0
      DO 210 J=J1,J2
C             INTERPOLATE FOR RA AND RB VERSUS XJ(J)
      DO 190 K=2,NFUSOR
      IF (XJ(J).LE.XFUS(K)) GO TO 200
  190 CONTINUE
      K = NFUSOR
C             INTERPOLATE BETWEEN VALUES
  200 DX = XFUS(K)-XFUS(K-1)
      IF (DX.NE.0.0) DX=(XJ(J)-XFUS(K-1))/DX
      RA2 = AMAX1(RA2,(FUSAZ(K)-FUSAZ(K-1))*DX+FUSAZ(K-1))
      RB2 = AMAX1(RB2,(FUSBY(K)-FUSBY(K-1))*DX+FUSBY(K-1))
  210 CONTINUE
  220 J1 = J2
      RETURN
      END
```

Figure C-1(ee)

```
      SUBROUTINE LAYBIP(NBDCR.BIL)
C
C  ROUTINE TO LAYOUT BODY INTERFERENCE PANELS USING SOURCE PANEL
C  GEOMETRY DATA.
C
      COMMON /RGEOM / XFUS(51).ZFUS(51).FUSARD(51).FUSRAD(51)
     *    .FUSAZ(51).XJ(51).PHIK(33)
      COMMON /CONSTS/ PI.DTOR
      COMMON /DIMENS/ IDIM(130)
      COMMON /ONE   / DELTP(250).FN(250).PNLC(250).SWPPLE(250).
     1   SWPPTE(250).XBAR(250).ZBAR(250).XCPT(250).YCPT(250).ZCPT(250)
     2   .XL (250).XLB(250).XRF(250).XRB(250).YLC(250).YRC(250).ZLF(250)
     3   .ZRF(250).ZLB(250).ZRB(250).SNT(100).CST(100).SNT2(100).
     4   CST2(250).XFBIP(100).ALFA.ALFR.B2.B2V.BETA.BETAR.CONST.
     5   CN.DX.EM.FMACH.SINALF.SINBET.SLOPE.TLRNC.
     6   TOTLR.U.V.W.UCHK.VCHK.WCHK.WBIP.X.Y.Z.IV.IF.II.JV.MSWR.MSWL.
     7   MSWU.MSWD.NBIP.NCW.NHP.NPR.NRP.N3P.NOCPT.NOUT.NPANLS.NWBP.
     8   RA.RB.ERATIO.BODY.DELTA
      COMMON/SWEEPS/VSWLER(20).VSWTER(20).VSWLEL(20).VSWTEL(20).
     1  VSWLEU(20).VSWTEU(20).VSWLED(20).VSWTED(20).LVSWP.LEFT.FAC.NCWB
     2.ARPNL(250).WIDTH(250)
      COMMON/WBTR/THTI(100).XWLE
      EQUIVALENCE  (IDIM(13).KRAD)
C
C
C     HERE K HAS ONE VALUE FOR EACH RING OF BIP*S
C     FOR EACH CIRCUMFERENTIAL RING. THE X FRONT POSITION. XFBIP(L)
C     IS A CONSTANT --
C     XFBIP(L) IS THE X-STATION OF THE BODY RING IN WING COORDINATES
C
      DX=BIL/FLOAT(NCWB)
      KX=0
      DO 230 K=1.NBIP.NBDCR
      XS=DX*FLOAT(KX)
      KP=K+NBDCR-1
      DO 220 L=K.KP
      XFBIP(L)=XS
  220 CONTINUE
      KX=KX+1
  230 CONTINUE
C
C     ********-**** LAY OUT BODY INTERFERENCE PANELS *******************
C
      IK = 0
      IPHI = 1
      YIM1 = 0.0
      ZIM1 = -RA
      KRAD2 = (KRAD+1)/2
      DO 260 I=1.NBDCR
C             DEFINE ANGLES ON RIGHT HALF OF INTERFERENCE SHELL
      IF (I.LT.KRAD2) IPHI=IPHI+1
      IF (I.LT.KRAD2) THTIDG = AMOD(PHIK(IPHI)/DTOR+270.0.360.)
C            DEFINE ANGLES ON LEFT HALF OF INTERFERENCE SHELL
      IF (I.GE.KRAD2) IPHI=IPHI-1
      IF (I.GE.KRAD2) THTIDG = 270.0-PHIK(IPHI)/DTOR
      THT = THTIDG*DTOR
      SN = SIN(THT)
      CS = COS(THT)
      RAD=1.0/SQRT((CS/RB)**2+(SN/RA)**2)
      YI = RAD*CS
      ZI = RAD*SN
      DY=YI-YIM1
      DZ=ZI-ZIM1
      WBIP=SQRT(DZ*DZ+DY*DY)
      SN2= DZ/WBIP
      CS2=-DY/WBIP
      AREAP=WBIP*DX
C
C        J IS THE PANEL INDEX OF BIPS
C        K IS THE PANEL INDEX OF WING AND BODY PANELS
C
      IK = IK+1
      DO 240 J=IK.NBIP.NBDCR
      K=J+NPANLS
      THTI(J)=THTIDG
      SNT(J)=SN
      CST(J)=CS
      SNT2(J)=SN2
      CST2(J)=CS2
C
C
      YCPT(K)=0.5*(YI+YIM1)
      ZCPT(K)=0.5*(ZI+ZIM1)
C
      SWPPLE(K)=0.0
      SWPPTE(K)=0.0
      ARPNL(K)=AREAP
C
C     NOTE -- IN THE **BIP** SYSTEM. YLC=0 AND YRC=WBIP
C
C     DEFINE BIP PANEL CORNER POINTS IN  WING COORDINATE SYSTEM
C
      YLC(K)=YI
      YRC(K)=YIM1
      ZLF(K)=ZI
      ZLB(K)=ZI
      ZRF(K)=ZIM1
      ZRB(K)=ZIM1
      XLF(K)=XFBIP(J)
      XRF(K)=XLF(K)
      XLB(K)=XLF(K)+DX
      XRB(K)=XLB(K)
C
C     CONTROL POINT X COORDINATE IN WING SYSTEM
C
      XBAR(K)=XFBIP(J)+0.5*DX
  240 XCPT(K)=XFBIP(J)+FAC*DX
      YIM1=YI
      ZIM1=ZI
  260 CONTINUE
      RETURN
      END


      SUBROUTINE LAYOUT (SLPWLE.SLPWTE.Y.MSWP.CRP.NS.CTP.PHI.THET)
C
C     VERSIONS DEMON2
C
C     THIS SUBROUTINE LAYS OUT AND DETERMINES GEOMETRICAL PROPERTIES
C     OF THE CONSTANT U-VELOCITY PANELS ON THE WING OR FIN SURFACES
C
      DIMENSION CSIDE(20).NPD(3).Y(1)
C
      LOGICAL LEFT.BODY.DELTA
C
      COMMON /CONSTS/ PI.DTOR
      COMMON /INTRDT/ PHIDIM.THETIT.PHIFR.PHIFL.PHIFU.PHIFD
      COMMON /ONE   / DELTP(250).FN(250).PNLC(250).SWPPLE(250).
     1   SWPPTE(250).XBAR(250).ZBAR(250).XCPT(250).YCPT(250).ZCPT(250)
     2   .XLF(250).XLB(250).XRF(250).XRB(250).YLC(250).YRC(250).ZLF(250)
     3   .ZRF(250).ZLB(250).ZRB(250).SNT(100).CST(100).SNT2(100).
     4   CST2(250).XFBIP(100).ALFA.ALFR.B2.B2V.BETA.BETAR.CONST.
     5   CN.DX.EM.FMACH.SINALF.SINBET.SLOPE.TLRNC.
     6   TOTLR.U.V.W.UCHK.VCHK.WCHK.WBIP.X.DY.Z.IV.IF.II.JV.MSWR.MSWL.
     7   MSWU.MSWD.NBIP.NCW.NHP.NPR.NRP.N3P.NOCPT.NOUT.NPANLS.NWBP.
     8   RA.RB.ERATIO.BODY.DELTA
      COMMON/SWEEPS/VSWLER(20).VSWTER(20).VSWLEL(20).VSWTEL(20).
     1  VSWLEU(20).VSWTEU(20).VSWLED(20).VSWTED(20).LVSWP.LEFT.FAC.NCWB
     2.ARPNL(250).WIDTH(250)
C
C
C     FUNCTION DEFINING THE RADIUS OF AN ELLIPSE IN TERMS OF THT
C
      FRAD(SN.CS)=1.0/SQRT((CS/RB)**2+(SN/RA)**2)
```

Figure C-1(ff)

```
      FYROT(YP) = COSINE*YP+YBOD
      FZROT(YP) = SINE  *YP+ZBOD
C
C  NS(=IL+1) IS WING QUADRANT INDICATOR
C      NS=1, RIGHT FIN-  =2, LEFT FIN-  =3, UPPER FIN-  =4, LOWER FIN
C     REFER TO MAIN PROGRAM CRFWBD FOR CORRESPONDENCE BETWEEN FINS ON
C     CRUCIFORM AND INTERDIGITATED CONFIGURATIONS.
C
      IF (NS.EQ.5) RETURN
      LEFT = NS.EQ.2 .OR. NS.EQ.4
      ANCW=NCW
      SLPDIF= SLPWLE-SLPWTE
      CSIDE(1)= CRP
      NPD(1)=NRP
      NPD(2)=NWP
      NPD(3)=N3P
C
C     VERTICAL WING TREATED IN THE SAME WAY AS HORIZONTAL WING
C     Y AND Z COORDINATES ARE INTERCHANGED FOR VERTICAL WING
C
C     LEFT STANDS FOR EITHER LEFT HORIZONTAL WING OR FOR LOWER
C     VERTICAL WING
C
C
C  LOCATE (Y,Z) OF WING BODY JUNCTION
C
      IF (THET.EQ.90.0) GO TO 100
      GO TO 101
  100 CS=0.0
      SN=1.0
      GO TO 104
  101 CONTINUE
      CS = COS(THET*DTOR)
      SN = SIN(THET*DTOR)
  104 CONTINUE
      RAD = FRAD(SN,CS)
      IF (LEFT) RAD=-RAD
      YBOD = RAD*CS
      ZBOD = RAD*SN
C
C        DEFINE DIHEDRAL ANGLES OF FINS
C
      IF (PHI.EQ.90.0) GO TO 102
      GO TO 103
  102 COSINE=0.0
      SINE=1.0
      GO TO 105
  103 CONTINUE
      COSINE = COS(PHI*DTOR)
      SINE   = SIN(PHI*DTOR)
  105 CONTINUE
C
C     INDEX I RUNS SPANWISE, K CHORDWISE ALONG WING
C     CALCULATE LENGTHS OF OUTBOARD PANEL SIDE,CSIDE
C     J IS THE PANEL NUMBER
C
      WLEX=0.0
      DO 140 I=2,NSWP
      IM=I-1
C
C     LVSWPB  VARIABLE WING SWEEP OPTION
C        LVSWP=0,  GENERATE  Y  INTERNALLY IN CRFWBD
C        LVSWP=1,  READ IN  Y*S.
C
      IF (LVSWP.EQ.0) GO TO 40
      GO TO (42,41,30,31),NS
C
C     CASE FOR YAWED WING, STREAMWISE PLANE PASSING THROUGH ROOTCHORD
C     LE INTERSECTS WING TE
C     TE SWEEPS MAY VARY AS A CONSEQUENCE
C
C     PANELS ON RIGHT HAND WING
C
   42 SLPWLE= TAN(VSWLER(I)*DTOR)
      SLPWTE= TAN(VSWTER(I)*DTOR)
      GO TO 44
C
C     PANELS ON LEFT HAND WING
C
   41 SLPWLE= TAN(-VSWLEL(I)*DTOR)
      SLPWTE= TAN(-VSWTEL(I)*DTOR)
      GO TO 44
C
C     PANELS ON UPPER WING
C
   30 SLPWLE=TAN(VSWLEU(I)*DTOR)
      SLPWTE=TAN(VSWTEU(I)*DTOR)
      GO TO 44
C
C     PANELS ON LOWER WING
C
   31 SLPWLE=TAN(-VSWLED(I)*DTOR)
      SLPWTE=TAN(-VSWTED(I)*DTOR)
C
C
   44 SLPDIF= SLPWLE-SLPWTE
      CSIDE(I)= CSIDE(IM)-(Y(I)-Y(IM))*SLPDIF
      IF (LEFT) CSIDE(I)=CSIDE(IM)+(Y(I)-Y(IM))*SLPDIF
      GO TO 43
C
C     NONVARYING L.E. AND T.E. SWEEPS  CONSTANT VALUES USED
C
   40 CSIDE(I)=CRP-(Y(I)-Y(1))*SLPDIF
      IF (LEFT) CSIDE(I)= CRP+(Y(I)-Y(1))*SLPDIF
   43 CONTINUE
C
C     SWPPLE AND SWPPTE ARE PANEL L.E. AND T.E. SWEEPS
C     CALCULATE PANEL CORNER POINT COORDINATES
C
      IF (I.EQ.2) GO TO 45
      JLE=(I-3)*NCW+1
      IF (NS.GT.1) JLE=JLE+NPD(NS-1)
      WLEX=XRF(JLE)
      IF (LEFT) WLEX=XLF(JLE)
   45 CONTINUE
C
      DO 130 K=1,NCW
      J=(I-2)*NCW+K
      IF (NS.GT.1) J=J+NPD(NS-1)
      AKM=K-1
      AK=K
      SWPPLE(J)=SLPWLE-AKM*SLPDIF/ANCW
      SWPPTE(J)=SLPWLE-AK*SLPDIF/ANCW
      IF (ABS(SWPPLE(J)).LE.0.001) SWPPLE(J)=0.0
      IF (ABS(SWPPTE(J)).LE.0.001) SWPPTE(J)=0.0
      IF (LEFT) GO TO 50
C
C  PANEL LAY OUT FOR RIGHT RIGHT HAND WING CORNER POINYS.
C     THEY APPLY TO RIGHT AND UPPER FINS.
C
      XLF(J)=AKM*CSIDE(IM)/ANCW+WLEX
      XLB(J)=XLF(J)+CSIDE(IM)/ANCW
      XRF(J)=AKM*CSIDE(I)/ANCW+(Y(I)-Y(IM))*SLPWLE+WLEX
      XRB(J)=XRF(J)+CSIDE(I)/ANCW
      YLC(J)=Y(IM)
      YRC(J)=Y(I)
      GO TO 51
C
C  PANEL LAY OUT FOR LEFT HAND WING CORNER POINTS.
C     THEY APPLY TO LEFT AND LOWER FINS.
C
   50 CONTINUE
      XLF(J)=AKM*CSIDE(I)/ANCW-(Y(I)-Y(IM))*SLPWLE+WLEX
      XRF(J)=AKM*CSIDE(IM)/ANCW+WLEX
      XRB(J)= XRF(J)+CSIDE(IM)/ANCW
      XLB(J)= XLF(J)+CSIDE(I)/ANCW
      YLC(J)=Y(I)
      YRC(J)=Y(IM)
```

Figure C-1(gg)

```
   51 CONTINUE
      ZLF(J)=0.0
      ZRF(J)=0.0
      ZLB(J)=0.0
      ZRB(J)=0.0
C
C     FIND Y-COORDINATE OF CONTROL POINT AND PANEL CENTROID
C
      A1=XRB(J)-XRF(J)
      A2=XLB(J)-XLF(J)
      H=Y(I)-Y(IM)
      YBAR=(2.0*A1+A2)*H/(3.0*(A1+A2))
      YCPT(J)=Y(IM)+YBAR
      IF (LEFT) YBAR=-(2.0*A2+A1)*H/(3.0*(A1+A2))
      IF (LEFT) YCPT(J)=Y(IM)-YBAR
C
C     FIND X-COORDINATE OF CONTROL POINT AND PANEL CENTROID
C
      XPCLE=XLF(J)+YBAR *SWPPLE(J)
      XPCTE=XLB(J)+YBAR*SWPPTE(J)
      IF (LEFT) XPCLE=XRF(J)+YBAR*SWPPLE(J)
      IF (LEFT) XPCTE=XRB(J)+YBAR*SWPPTE(J)
      PNLC(J)=XPCTE-XPCLE
      XCPT(J)=XPCLE+FAC *PNLC(J)
      XBAR(J)=(XPCTE+XPCLE)/2.0
C
C
      ZCPT(J)=0.0
      ZBAR(J)=0.0
      IF (.NOT.LEFT) GO TO 120
      IF (SWPPLE(J).EQ.0.0) SWPPLE(J)=TLRNC
      IF (SWPPTE(J).EQ.0.0) SWPPTF(J)=TLRNC
      SWPPLE(J)=-SWPPLE(J)
      SWPPTE(J)=-SWPPTE(J)
  120 CONTINUE
C
C        AREA AND WIDTH OF PANEL IN LOCAL COORDINATES
C
  121 ARPNL(J)=0.5*(YRC(J)-YLC(J))*(XLB(J)-XLF(J)+XRB(J)-XRF(J))
      WIDTH(J)=YRC(J)-YLC(J)
C
C     TRANSFORM FROM FIN COORDINATES TO THE WING REFERENCE COORDINATE
C     SYSTEM XW,YW,ZW.
C
      ZLF(J) = FZROT(YLC(J)-Y(1))
      ZLB(J) = ZLF(J)
      YLC(J) = FYROT(YLC(J)-Y(1))
      ZRF(J) = FZROT(YRC(J)-Y(1))
      ZRB(J) = ZRF(J)
      YRC(J) = FYROT(YRC(J)-Y(1))
      ZBAR(J) = FZROT(YCPT(J)-Y(1))
      ZCPT(J) = ZBAR(J)
      YCPT(J) = FYROT(YCPT(J)-Y(1))
  130 CONTINUE
  140 CONTINUE
      CTP=CSIDE(MSWP)
      RETURN
      END


      SUBROUTINE LOADS (HEAD)
C
C     VERSION$DEMON2.
C
C     THIS SUBROUTINE CALCULATES FORCES AND MOMENTS ACTING ON THE
C     WINGS OR FINS AND THE INTERFERENCE SHELL.
C     THEY ARE FIRST CALCULATED IN THE WING REFERENCE COORDINATE SYSTEM
C     AND THEN TRANSFORMED TO WIND-AXIS SYSTEM.
C
C     A DESCRIPTION OF THE EQUATIONS USED TO COMPUTE THE LOADS FOR
C     INTERDIGITATED AND CRUCIFORM FINS IS FOUND IN APPENDIX F
C     IN NASA CR-3122.   A DESCRIPTION OF THE EQUATIONS USED TO
C     COMPUTE THE NORMAL FORCE CALCULATED ON AN AN ELLIPTIC BODY
C     MAY BE FOUND IN SECTION 4.2  OF  NASA CR-3122.
C
      DIMENSION BFY(100),BFZ(100),HEAD(20),MSW(4),A  LE(4),PHI(4)
     *    ,CZFIN(4),CYFIN(4),CMFIN(4),CNFIN(4),CLFIN(4)
C
      LOGICAL DELTA
C
      COMMON /CONSTS/ PI,DTOR
      COMMON /INTHDT/ PHIDIH,THETIT,PHIFR,PHIFL,PHIFU,PHIFD
      COMMON/OAFM/XM,ZM,CZOA,CYOA,CMOA,CLNOA,CLLOA
      COMMON /ONE    / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLB(250),XRF(250),XRB(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLB(250),ZRB(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLH,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
     7   MSWU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
      COMMON/SPCPRS/DLTPG(250)
      COMMON/SWEEPS/VSWLER(20),VSWTER(20),VSWLEL(20),VSWTEL(20),
     1  VSWLEU(20),VSWTEU(20),VSWLED(20),VSWTED(20),LVSWP,LEFT,FAC,NCWB
     2,ARPNL(250),WIDTH(250)
      COMMON/THREE/ANGLR,ANGLL,ANGLU,ANGLD,DELR,DELL,DELU,DELD,SREF,REFL
      COMMON/WBTR/THTI(100),XWLE
      EQUIVALENCE (MSW(1),MSWR),(ANGLE(1),ANGLR),(PHI(1),PHIFR)
C
C
C     INITIALIZE VARIABLES
C
      CNB=0.
      CYB=0.
      CZBIP=0.0
      CYBIP=0.0
      CMBIP=0.0
      CLNBIP=0.0
      CLLBIP=0.0
C
C
      AREA=0.0
      CZOA=0.0
      CYOA=0.0
      CMOA=0.0
      CLNOA=0.0
      CLLOA=0.0
      DO 100 L=1,4
      CZFIN(L)=0.0
      CYFIN(L)=0.0
      CMFIN(L)=0.0
      CLFIN(L)=0.0
      CNFIN(L)=0.0
  100 CONTINUE
C
C     PUT MOMENT CENTER IN WING COORDINATE SYSTEM.
C
      XMW=XM-XWLE
C
C
C
C     LOADS ON BODY OVER INTERFERENCE SHELL LENGTH
C     PRESSURES  (DLTPG) ACCOUNT FOR BODY, FIN AND INTERFERENCE SHELL
C     EFFECTS
C
      DO 140 K=1,NBIP
      I=K+NPANLS
      F=-DLTPG(I)*ARPNL(I)
      FN(I)=F
      CALL ROTBW(0.,F,BFY(K),BFZ(K),K)
      CNB=CNB+BFZ(K)
      CYB=CYB+BFY(K)
      ARM=XBAR(I)-XMW
      BFZM=-BFZ(K)*ARM
      CMBIP=CMBIP+BFZM
```

Figure C-1(hh)

```
      BFYM=-BFY(K)*ARM
      CLNBIP=CLNBIP+BFYM
      BIPRM=-BFZ(K)*YCPT(I)+BFY(K)*(ZCPT(I)-ZM)
      CLLBIP=CLLBIP+BIPRM
  140 CONTINUE
      CZBIP=CNB/SREF
      CYBIP=CYB/SREF
      CMBIP=CMBIP/(SREF*REFL)
      CLNBIP=CLNBIP/(SREF*REFL)
      CLLBIP=CLLBIP/(SREF*REFL)
C
C
C
C     CALCULATE FORCES,CZFIN,CYFIN, ACTING ON THE FINS.
C     PITCHING MOMENT,CMFIN,YAWING MOMENT,CLNF,ROLLIN MOMENT,CLLF.
C     FORCE. LAST LETTER DESIGNATES FIN AS FOLLOWS.
C      1-R.....HOR. RIGHT FIN
C      2-L.....HOR. LEFT FIN
C      3-U.....VERT. UPPER FIN
C      4-D.....VERT. LOWER FIN
C     BIP......INTERFERENCE SHELL.
C
C     ALSO%
C     IN WING OR BODY COORDINATE SYSTEM.
C     FX.....THRUST FORCE IN NEG. X-DIR. IN PLANE OF WING/Q
C     FY.....SIDE FORCE IN Y-DIRECTION IN PLANE OF WING/Q
C     FZ.....UPWARDS FORCE IN Z-DIR. IN PLANE OF VERTICAL WING
C  NOTE%  ADDITIONAL CONTRIBUTIONS TO IN PLANE FORCES ARE CALCULATED
C     IN SUBROUTINE SPNLD.
C     PRESENTLY, IN-PLANE FORCES  SUITABLE FOR CRUCIFORM OR PLANAR
C     FINS OR WINGS ONLY.
C
C     THE 300 LOOP IS USED FOR THE RIGHT AND LEFT HORIZONTAL PANELS
C
      JU = 0
      DO 310 L=1,4
      IF (MSW(L).EQ.0) GO TO 310
      JL = JU+1
      JU = JU+MSW(L)*NCW
      PHIF=PHI(L)*DTOR
      DO 300 J=JL,JU
      AREA=AREA+ARPNL(J)
      F=    ARPNL(J)*DLTPG(J)
      FN(J)=F
      ARM=XBAR(J)-XMW
      CALL ROTFW(0.0,F,CYP,CZP,PHIF)
      CMP=-CZP*ARM
      CLNP=-CYP*ARM
      PRM=-CZP*YCPT(J)+CYP*(ZCPT(J)-ZM)
      CZFIN(L)=CZFIN(L)+CZP
      CYFIN(L)=CYFIN(L)+CYP
      CMFIN(L)=CMFIN(L)+CMP
      CNFIN(L)=CNFIN(L)+CLNP
      CLFIN(L)=CLFIN(L)+PRM
      SLPMC=0.5*(SWPPLE(J)+SWPPTE(J))
  300 CONTINUE
C
      CZFIN(L)=CZFIN(L)/SREF
      CYFIN(L)=CYFIN(L)/SREF
      CMFIN(L)=CMFIN(L)/(SREF*REFL)
      CNFIN(L)=CNFIN(L)/(SREF*REFL)
      CLFIN(L)=CLFIN(L)/(SREF*REFL)
310   CONTINUE
C
C
C     OVERALL FORCE AND MOMENT COEFFICIENTS FOR FINS AND INTERFERENCE
C     SHELL IN WING OR BODY REFERENCE SYSTEM.
C
C     CYOA.......ACTS ALONG POSITIVE Y-AXIS
C     CZOA.......ACTS ALONG POSITIVE Z-AXIS
C     CMOA.......VECTOR ALONG NEGATIVE Y-AXIS,NOSE UP POS.
C     CLNOA......VECTOR ALONG POSITIVE Z-AXIS,NOSE TO RIGHT POS.
C     CLLNOA.....VECTOR ALONG POSITIVE X-AXIS, RIGHT WING DOWN POS.
C
C     FORCE COEFFICIENTS IN X,Y,Z DIRECTIONS.
C
      CZOA=CZFIN(1)+CZFIN(2)+CZFIN(3)+CZFIN(4)+CZBIP
      CYOA=CYFIN(1)+CYFIN(2)+CYFIN(3)+CYFIN(4)+CYBIP
C
C     PITCHING,YAWING,ROLLING MOMENTS COEFFICIENTS.
C
      CMOA=CMFIN(1)+CMFIN(2)+CMFIN(3)+CMFIN(4)+CMBIP
      CLNOA=CNFIN(1)+CNFIN(2)+CNFIN(3)+CNFIN(4)+CLNBIP
      CLLOA=CLFIN(1)+CLFIN(2)+CLFIN(3)+CLFIN(4)+CLLBIP
C
C
C
C
      IF (THETIT.NE.0.0) GO TO 999
      IF (MSWR.GT.0 .AND. PHIFR.NE.0.0) GO TO 999
      IF (MSWL.GT.0 .AND. PHIFL.NE.0.0) GO TO 999
      IF (MSWU.GT.0 .AND. PHIFU.NE.90.) GO TO 999
      IF (MSWD.GT.0 .AND. PHIFD.NE.90.) GO TO 999
C
C  COMPUTE SPANLOAD DISTRIBUTION AND NET VORTICITY AT TRAILING
C  EDGE OF FINS
C
      CALL SPNLD
999   RETURN
      END




      SUBROUTINE NUMACH (Y,Z)
C
C     THIS SUBROUTINE DETERMINES THE LEADING EDGE AND TRAILING EDGE
C          LOCATIONS AND MACH NUMBERS ASSOCIATED WITH WING THICKNESS
C
C     NOTE% X,Y,Z IN WING COORDINATE SYSTEM
C
      COMMON /CONFIG/NFU,NPY,NSTHS,NRACK
      COMMON /FLOW/DUM6(2),FMACH,DUM7(2),BETA,DUM8
      COMMON /FSOR/FXL(101),FSS(100),FDS(100),NFSOR
      COMMON/GRIDCU/XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
     1      SWP(500),SNBP(40),CSBP(40),THTBP(40),DPNET(500)
      COMMON /LETEM/ FMXT3,FMXT4
      COMMON/NUFLOW/FNUMCH,BETANU,BTSONU,INUMCH,ISMTH,XT3,XT4
      COMMON/NUINDX/NRW,NRP,NRB,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1      NCW1,NCP1,NCWB1,NCWS,NCWS1,NCPS1
      COMMON/VELARG/DUM5(9),UP,VP,WP
      COMMON/WGEOM/XBWOC,ZBWO,CRW,ZOIMED
C
  500 FORMAT(//5X58H*****  AXIAL LOCATION OF TRAVERSE ABOVE WING CHORDAL
     1 PLANE,47X,12H** NUMACH **)
C
C     CALCULATF LOCAL WING LE X COORDINATE AND Z COORDINATE
C
      YY=Y
      IF(YY.GT.0.) YY=-YY
      IF(YY.GE.YPT(1)) GO TO 15
      DO 10  I=2,NRW
      JLER=(I-2)*NCW1+1
      IF(YY.GE.YPT(I)) GO TO 20
   10 CONTINUE
C
C     YY LESS THAN Y OF WING TIP. USE TIP
C     CHORD FOR CHHD.
C
      JLEL=JLER+NCW1
      WLEX=XPT(JLEL)
      WTEX=XPT(NPTW)
      ZLOC=ZPT(NRW)
      GO TO 25
C
C     YY GE RIGHT SIDE OF FIRST CHORDWISE ROW.
C     USE ROOT-CHORD FOR CHRD.
C
```

Figure C-1(ii)

```
   15 WLEX=0.0
      WTEX=-CRW
      ZLOC=0.0
      GO TO 25
C
C     YY BRACKETED.  INTERPOLATE AND COMPUTE CHRD.
C
   20 DELT=YY-YPT(I-1)
      WLEX=XPT(JLER)+DELT*SWP(JLER)
      JTE=JLER+NC-1-1
      WTEX=XPT(JTE)+DELT*SWP(JTE)
      ZLOC=ZPT(I-1)+DELT*SPHI(I-1)/CPHI(I-1)
   25 CHRD=WLEX-WTEX
      INUMCH=0
C
      IF(Z.GT.ZLOC) GO TO 30
      WRITE(6,500)
      STOP
C
C     HUNT FOR BEGINNING OF WING THICKNESS VELOCITY FIELD. FIRST USING
C          THE FREE-STREAM MACH NUMBER AND THEN USING THE CALCULATED
C          LOCAL MACH NUMBER
C
   30 DELX1=0.01*CHRD
      TSAVE=0.0
      X=0.0
      XSAVE=0.0
C
C     SEARCH FOR MAXIMUM TURNING ANGLE USING DELTA X=0.01*CHRD
C
  101 X=X-DELX1
C
C     THE FOLLOWING TEST IS APPROXIMATE.  IF NO FIRST INFLUENCE
C     PROPOGATING FROM ABOUT TWO CHORD LENGTHS BEHIND THE WING THE
C     FOLLOWING SEARCH IS TERMINATED
C
      TEST=X+2.0*CHRD-WLEX+BETA*Z
      IF (TEST.LE.0.0) STOP 101
      CALL VELWT2(X,Y,Z)
      WP=-WP
      UP=1.0-UP
      DNU=SQRT(VP*VP+WP*WP)/UP
      DNU=ATAN(DNU)*57.2957795
      IF (DNU.LE.0.0) GO TO 101
      IF (DNU.LE.(0.8*TSAVE)) GO TO 102
      IF (DNU.LE.TSAVE) GO TO 101
      TSAVE=DNU
      XSAVE=X
      WSAVE=WP
      GO TO 101
C
C     BACK UP TO XSAVE-DELX1 AND SEARCH FOR MAXIMUM TURNING ANGLE
C          USING DELTA X=0.001*CHRD
C
  102 X=XSAVE+DELX1
      XFIN=XSAVE-DELX1
      DELX1=0.1*DELX1
  104 X=X-DELX1
      CALL VELWT2(X,Y,Z)
      WP=-WP
      UP=1.0-UP
      DNU=SQRT(VP*VP+WP*WP)/UP
      DNU=ATAN(DNU)*57.2957795
      IF (X.LE.XSAVE) GO TO 106
      IF (DNU.GE.TSAVE) GO TO 105
      GO TO 103
  106 IF (DNU.LE.TSAVE) GO TO 103
  105 TSAVE=DNU
      WSAVE=WP
      XSAVE=X
  103 IF (X.GT.XFIN) GO TO 104
C
C     CALCULATE FLOW ANGLE AT (XT3,Y,Z) POSITION IF INUMCH=0
C
      XT3=XSAVE
      ANGNU1=2.4495*ATAN(0.40825*BETA)-ATAN(BETA)
      ANGNU1=ANGNU1*57.2957795
      TSAVE=ATAN(ABS(WSAVE))*57.2957795
      ANGNU2=ANGNU1-TSAVE
C
C     CALCULATE LOCAL MACH NUMBER DUE TO ISENTROPIC COMPRESSION AT
C     (XT3,Y,Z) POSITION USING I.M.HALL METHOD (AERO. JOUR. SEPT 75)
C     NOTE$THIS NEW MACH NUMBER IS TAKEN IN X DIRECTION
C
      IF (ANGNU2.GT.0.1) GO TO 110
  112 FNUMCH=1.01
      GO TO 111
  110 YNU=(ANGNU2/130.4540769)**0.6666666667
      YNUSQ=YNU*YNU
      YNUC=YNU*YNUSQ
      FNUMCH=((1.0+1.3604*YNU+0.0962*YNUSQ-0.5127*YNUC)/(1.0-0.6722*YNU
     1 -0.3278*YNUSQ))*COS(TSAVE/57.2957795)
      IF (FNUMCH.LT.1.01) GO TO 112
  111 INUMCH=1
      BTSQNU=FNUMCH*FNUMCH -1.0
      BETANU=SQRT(BTSQNU)
      XT3SVE=XT3
C
C     REPEAT SEARCH FOR XT3 USING NEW MACH NUMBER
C
      ITRIP=0
      DELX1=0.01*CHRD
      X=0.0
      XSAVE=0.0
  150 X=X-DELX1
      CALL VELWT2 (X,Y,Z)
      XSAVE=X
      IF (WP.EQ.0.0) GO TO 150
      IF (ITRIP.NE.0) GO TO 160
C
C     BACK UP ONE STEP AND REPEAT SEARCH USING DELTA X=0.001*CHRD
C
      ITRIP=1
      X=XSAVE+DELX1
      XSAVE=X
      DELX1=0.1*DELX1
      GO TO 150
  160 XT3=XSAVE
C
C     DEFINE XT3 IN FUSELAGE COORDINATE SYSTEM AND SET FMXT3 EQUAL
C          TO LOCAL MACH NUMBER AT THIS POINT
C
      XT3=XT3+XBWOC
      FMXT3=FNUMCH
C
C     DETERMINE DOWNSTREAM POINT WHERE W VELOCITY DUE TO WING THICKNESS
C          JUMPS BACK TO A VALUE NEAR ZERO. FIRST USING THE FREE-STREAM
C          MACH NUMBER AND THEN USING THE LOCAL MACH NUMBER
C
      INUMCH=0
  203 DELX1=0.1*CHRD
      X=XT3SVE-0.7*CHRD
      CALL VELWT2(X,Y,Z)
      WP=-WP
      XSAVE=X
      WSAVE=WP
C
C     SEARCH FOR JUMP USING DELTA X = 0.1*CHRD
C
  201 X=X-DELX1
      CALL VELWT2(X,Y,Z)
      WP=-WP
      DELTW=WSAVE-WP
      IF (DELTW.GE.(0.5*WSAVE)) GO TO 210
      IF (WP.LT.WSAVE) GO TO 201
      WSAVE=WP
      XSAVE=X
      GO TO 201
```

Figure C-1(jj)

```
C
C     BACK UP TO XSAVE-DELX1 AND REPEAT SEARCH USING DELTA X = 0.01*CHRD
C
  210 IF (INUMCH.EQ.1) GO TO 230
      X=XSAVE+DELX1
      XFIN=XSAVE-DELX1
      DELX1=0.1*DELX1
  211 X=X-DELX1
      CALL VELWT2(X,Y,Z)
      WP=-WP
      IF (X.LE.XSAVE) GO TO 214
      IF (WP.GE.WSAVE) GO TO 212
      GO TO 213
  214 IF (WP.LT.WSAVE) GO TO 213
  212 WSAVE=WP
      XSAVE=X
  213 IF (X.GT.XFIN) GO TO 211
C
C     CALCULATE FLOW ANGLE TNU AT (XT4,Y,Z) POSITION IF INUMCH=0
C
      XT4=XSAVE
      CALL VELWT2(XT4,Y,Z)
      WP=-WP
      UP=1.0-UP
      TANNU=SQRT(VP*VP+WP*WP)/UP
      TNU=-57.2957795*ATAN(TANNU)
      ANGNU2=ANGNU1-TNU
C
C     CALCULATE LOCAL MACH NUMBER AT (XT4,Y,Z)
C
      YNU=(ANGNU2/130.4540769)**0.6666666667
      YNUSQ=YNU*YNU
      YNUC=YNU*YNUSQ
      FNUMCH=((1.0+1.3604*YNU+0.0962*YNUSQ-0.5127*YNUC)/(1.0-0.6722*YNU
     1 -0.3278*YNUSQ))*COS(TNU/57.2957795)
      INUMCH=1
      BTSQNU=FNUMCH*FNUMCH -1.0
      BETANU=SQRT(BTSQNU)
C
C     GO BACK AND REPEAT SEARCH FOR XT4 USING NEW MACH NUMBER
C
      GO TO 203
C
C     WHEN INUMCH=1 SEARCH FOR DROP IN W VELOCITY AND USE THIS LOCATION
C          FOR XT4
C
  230 XFIN=X
      X=X+DELX1
      DELX1=0.1*DELX1
  231 X=X-DELX1
      CALL VELWT2(X,Y,Z)
      WP=-WP
      DELTW=WSAVE-WP
      IF (DELTW.GE.(0.5*WSAVE)) GO TO 232
      IF (X.GT.XFIN) GO TO 231
  232 XT4=X+DELX1
      XT4=XT4+XBWOC
      FMXT4=FNUMCH
      RETURN
      END


      SUBROUTINE PANVEL (UB,VB,WB,AN,IXZSYM)
C
C ROUTINE TO ORGANIZE THE CALCULATION OF THE INFLUENCE OF A GIVEN
C PANEL *J* ON A FIELD POINT (XPTI,YPTI,ZPTI).  UB,VB,WB ARE THE
C THREE ORTHOGONAL VELOCITY INFLUENCES ASSOCIATED WITH THE BODY AXES.
C *AN* IS THE NORMAL INFLUENCE.
C
      LOGICAL LZERO,SUBSON,SUPERS
      COMMON /BLKPAN/ COST,SINT,XBTJ,YBTJ,ZBTJ,XC1,YC1,ZC1
     *    ,XPTI,YPTI,ZPTI,COSTI,SINTI,COSD,SINO,LZERO
      COMMON /BODCOM/ AMACH,TANO,CX,XCOR(4),YCOR(4),ZCOR(4)
     1    ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,J
C
C              LZERO = .FALSE. - DEFINE ANGLE TRANSFORMATIONS
C                      .TRUE.  - SKIP TRANSFORMATION CALCULATION
      IF (LZERO) GO TO 40
      XW=SINT*COSTI
      XX=COST*SINTI
      XY=COST*COSTI
      XZ=SINT*SINTI
      SINTR=XW-XX
      SINTL=XW+XX
      COSTR=XY+XZ
      COSTL=XY-XZ
C
C CALCULATION OF CONTROL POINT IN PANEL COORDINATE SYSTEM
C
   40 XI=XPTI-XC1
      DY=YPTI-YC1
      DZ=ZPTI-ZC1
      YI=DY*COST+DZ*SINT
      ZI=DZ*COST-DY*SINT
      XJ =XBTJ-XC1
      DYJ=YBTJ-YC1
      DZJ=ZBTJ-ZC1
      ZJ=DZJ*COST-DYJ*SINT
C
C CALCULATE VELOCITY COMPONENTS INDUCED BY CONSTANT SOURCE
C DISTRIBUTION PANELS
C
      CALL SORPAN(UR,VR,WR)
      UB=UR
      VI=VR*COSTR-WR*SINTR
      WI=VR*SINTR+WR*COSTR
      UL=0.
      VL=0.
      WL=0.
C
C CALCULATE VELOCITY COMPONENTS FROM LEFT SIDE FOR SYMMETRIC
C CONFIGURATIONS INDUCED BY CONSTANT SOURCE PANEL
C
      IF (IXZSYM.NE.0) GO TO 45
      DY=-YPTI-YC1
      YI=DY*COST+DZ*SINT
      ZI=DZ*COST-DY*SINT
      CALL SORPAN(UL,VL,WL)
      UB=UB+UL
      VI=VI-VL*COSTL+WL*SINTL
      WI=WI+VL*SINTL+WL*COSTL
   45 CONTINUE
C
C CALCULATE VELOCITY COMPONENTS IN ORIGINAL COORDINATE SYSTEM
C
      VB=VI*COSTI-WI*SINTI
      WB=WI*COSTI+VI*SINTI
      AN=WI*COSD-UB*SINO
      RETURN
      END


      SUBROUTINE PAS001 (A,N,NER,ND)
C
C LU DECOMPOSITION OF POSITIVE DEFINITE MATRIX - A
C ARGUEMENTS&
C   A   = MATRIX OF COEFFICIENTS, A(ND,N)
C   N   = ACTUAL SIZE OF A(N,N)
C   NER = DIAGNOSTIC.  IF NER.GT.0 ON RETURN, DECOMPOSITION FAILED
C           BECAUSE ABS(A(NER,NER)) ELEMENT IS LESS THAN 1.0E-20
C   ND  = DIMENSIONED SIZE OF A.
C UPON RETURN, A CONTAINS LU WHERE THE DIAGONAL ELEMENTS OF L ARE 1.
C BY G.N. VANDERPLAATS                            JAN. 1973
C NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CAL
```

Figure C-1(kk)

```
C
      DIMENSION A(ND,N)
CDC   DOUBLE PRECISION A,AA,BB
      NM1=N-1
      NER=0
      DO 40 K=1,NM1
      KP1=K+1
      AA=A(K,K)
      IF (ABS(AA).GT.1.0E-20) GO TO 10
C  SINGULAR MATRIX
      NER=K
      RETURN
10    AA=1.0/AA
      DO 30 I=KP1,N
      BB=-A(I,K)*AA
      IF (BB.EQ.0.0) GO TO 30
      A(I,K)=BB
      DO 20 J=KP1,N
20    A(I,J)=A(I,J)+BB*A(K,J)
30    CONTINUE
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PAS002 (A,B,N,NB,NDA,NDB)
C
C  SOLVES THE SYSTEM OF EQUATIONS LU*X = B, BY FORWARD AND BACKWARD
C  SUBSTITUTION.  IT IS ASSUMED THAT MATRIX, A, HAS BEEN DECOMPOSED
C  BY ROUTINE PAS001 OR EQUIVALENT SO THAT A CONTAINS LU.
C  ARGUEMENTS$
C    A    = COEFFICIENT MATRIX CONTAINING LU.  A(NDA,N)
C    B    = MATRIX CONTAINING RIGHT HAND SIDES OF THE LINEAR SYSTEM
C            LU*X = B
C    N    = ACTUAL SIZE OF LU MATRIX CONTAINED IN A.
C    NB   = NUMBER OF RIGHT HAND SIDE VECTORS CONTAINED IN THE FIRST
C            NB COLUMNS OF B.
C    NDA = DIMENSIONED SIZE OF A.
C    NDB = DIMENSIONED SIZE OF B.
C  BY G.N. VANDERPLAATS                               JAN., 1973
C  NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CAL
C
      DIMENSION A(NDA,N),B(NDB,NB)
C     DOUBLE PRECISION A,B,AA
      NM1=N-1
      NP1=N+1
C
C  FORWARD SUBSTITUTION
      DO 10 K=1,NM1
      KP1=K+1
      DO 10 I=KP1,N
      AA=A(I,K)
      DO 10 J=1,NB
10    B(I,J)=B(I,J)+AA*B(K,J)
C
C  BACKWARD SUBSTITUTION
      AA=1.0/A(N,N)
      DO 20 J=1,NB
20    B(N,J)=AA*B(N,J)
      DO 50 K=2,N
      I=NP1-K
      J1=I+1
      DO 30 J=J1,N
      AA=A(I,J)
      DO 30 L=1,NB
30    B(I,L)=B(I,L)-AA*B(J,L)
      AA=1.0/A(I,I)
      DO 40 L=1,NB
40    B(I,L)=AA*B(I,L)
50    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FITHOL(XLOC,VXAV,WXAV,X,Y,VS,WS)
C
C  THIS ROUTINE COMPUTES THE CROSSFLOW COMPONENTS FOR AN ELLIPTIC BODY
C  SUBJECTED TO A NONUNIFORM FLOW FIELD VAV(X),WAV(X) AT CENTERLINE
C
      COMMON/BLK1/BY,AZ,R,S,PI,CI
      COMMON /WOY1  / VAV(50),WAV(50),XAV(50),NRING
      COMPLEX SZ,CI,ZA,SQ,SQS,VELAL,Z,DSDZ
      EXTERNAL Z,DSDZ
C
C  FIND INDICES IN TABLE FOR X-LOCATION
C
      DO 100 I=2,NRING
      IF (XLOC.LE.XAV(I)) GO TO 120
100   CONTINUE
      I = NRING
C
C  INTERPOLATE BETWEEN RINGS TO FIND AVERAGE VELOCITIES
C
120   IM1 = I-1
      DX = XAV(I)-XAV(IM1)
      IF (DX.NE.0.) DX=(XLOC-XAV(IM1))/DX
      VXAV = VAV(IM1)+(VAV(I)-VAV(IM1))*DX
      WXAV = WAV(IM1)+(WAV(I)-WAV(IM1))*DX
      IF ((VXAV.EQ.0.0).AND.(WXAV.EQ.0.0)) GO TO 999
C
C  CALCULATE CROSSFLOW VELOCITY COMPONENTS
C
      RR=R*R
      ZA=CMPLX(X,Y)
      SZ=Z(ZA)
      SQ=SZ*SZ
      SQS=DSDZ(ZA)
      VELAL=CMPLX(0.0,0.0)
      IF (WXAV.NE.0.0) VELAL=-CI*WXAV*(1.0+RR/SQ)*SQS+VELAL
      IF (VXAV.NE.0.0) VELAL=VELAL-VXAV*(1.0-RR/SQ)*SQS
      VS=REAL(VELAL)
      WS=-AIMAG(VELAL)
999   RETURN
      END
```

```
      SUBROUTINE PRESS(NBODY,U,V,W,CPP,CPSTAG,CPCRIT,CPVAC)
C
C  THE PRESSURE COEFFICIENT IS CALCULATED USING THE BERNOULLI PRESSURE
C  FORMULA LIMITED BY THE VACUUM PRESSURE COEFFICIENT
C
C  COMPUTE THE STAGNATION PRESSURE COEFFICIENT, CRITICAL PRESSURE
C  COEFFICIENT, AND VACUUM PRESSURE COEFFICIENT.
C
      COMMON /PARAM / XMACH,ALPHA,BETA,ALPHAC,PHIR,EM
     1    ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      DIMENSION U(1),V(1),W(1),CPP(1)
      XM2=XMACH*XMACH
      CPCRIT=0.
      CPSTAG=1.
      CPVAC=0.
      IF (XM2.EQ.0.) GO TO 10
      CON=1.42857/XM2
      CPVAC=-CON
      CON1=.2*XM2
C
C  LOOP THROUGH NBODY-PANELS TO CALCULATE PRESSURE COEFFICIENTS  -------
C
10    DO 40 J=1,NBODY
C              SQUARE OF TOTAL RESULTANT VELOCITY
      Q2 = 1.0+2.*(U(J)*COSAC-V(J)*SINB+W(J)*SINA)
     1    +U(J)*U(J)+V(J)*V(J)+W(J)*W(J)
```

Figure C-1(11)

```
      IF (XMACH.EQ.0.) GO TO 30
      ARG=1.+CON1*(1.-Q2)
      IF (ARG.LT.0.) ARG=0.
      CPP(J)=CON*(ARG**3.5-1.)
      GO TO 40
30    CPP(J)=1.-Q2
40    CONTINUE
      IF (XMACH.EQ.0.) RETURN
      CPSTAG=CON*((1.+CON1)**3.5-1.)
      CPCRIT=CON*((5./6.+XM2/6.)**3.5-1.)
      RETURN
      END


      SUBROUTINE RDFILE
C
C     THIS SUBROUTINE READS THE FILE PRODUCED BY PROGRAM 1 WHICH
C          CONTAINS ALL OF THE DATA READ INTO PROGRAM 1 OR COMPUTED
C          IN THAT PROGRAM WHICH IS USED BY THIS PROGRAM
C
C    COMMON STATEMENTS
C
      COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7),
     1        SSUMK(7),SSUMKD(7)
      COMMON /CONFIG/ NFU,NPY,NSTRS,NRACK
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101),
     1     FRMAX,FLTHC,NFPOLY,FXEND(7),FCOEF(7,7)
      COMMON /FSOR/ FXL(101),FSS(100),FDS(100),NFSOR
      COMMON/GRIDCU/XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
     1     SWP(500),SNBP(40),CSBP(40),THTBP(40),DPNET(500)
      COMMON/GRIDTH/XPTS(1000),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
     1     SWPS(1000),THTNET(1000),DZDX(400)
      COMMON /MULSTR/ NFSHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1  ,IDEJ,IDF,ID2
      COMMON/NUINDX/NRW,NRP,NRB,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1     NCW1,NCP1,NCWB1,NCWS,NCWS1,NCPS1
      COMMON /PYGEOM/ YPL,CENTER
      COMMON /RKGEOM/ RLTHC,RRMAX,RIBCR,CRIBCR,SRIBCR,YBRO,XBRO,ZBRO
      COMMON /RSHOCK/ NRSHK,RXSHK(50),RRSHK(50),RDRDX(101)
      COMMON /RSOR/ NRSOR,RXL(101),RSS(100),RDS(100)
      COMMON /SSHOCK/ NSSHK(7),SXSHK(50,7),SRSHK(50,7),SDRDX(101,7)
      COMMON /SSOR/ SXL(101,7),SSS(100,7),SDS(100,7),NSSOR(7)
      COMMON /STGEOM/ SLTHC(7),SRMAX(7),SIBCR(7),CSIBCR(7),SSIBCR(7),
     1               YBSO(7),XBSO(7),ZBSO(7),NUMSTR(7),NSHAPE(7)
     3               ,NSHPT,MSHAPE(7),SPHIRR(7)
      COMMON /VELARG/ X,Y,Z,U,V,W,EM,TLRNC,PYPNL,UP,VP,WP
      COMMON /WGEOM / XBWOC,ZBWO,CRW,ZDIHED
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
      REWIND 10
      REWIND 11
C
C     READ INDICES
C
      READ(12) NFU,NPY,NSTRS,NRW,NRP,NRB,NPTW,NRWS,NRPS,NPTWS,NRWP,
     1     NPTWP,NCW1,NCWS,NCWS1,NRTOT,NPTT,NRTOTS,NPTTS,NTHP,NRACK
C
C     READ  CONSTANTS
C
      READ  (12) ALFACR,FMACH,TLRNC,XBWOC,ZBWO,CRW,ZDIHED,BETA,BETASQ
C
C     READ  U VELOCITY PANEL DATA
C
      READ(12) (XPT(I),DPNET(I),I=1,NPTT)
      READ(12) (SWP(I),I=1,NPTWP)
      READ(12) (YPT(I),ZPT(I),I=1,NRTOT)
      READ(12) (SPHI(I),CPHI(I),I=1,NRW)
C
C     READ  THICKNESS PANEL DATA
C
      READ(12) (XPTS(I),SWPS(I),THTNET(I),I=1,NPTTS)
      READ(12) (YPTS(I),ZPTS(I),I=1,NRTOTS)
      READ(12) (SPHS(I),CPHS(I),I=1,NRWS)
      READ(12) (DZDX(I),I=1,NTHP)
C
C     READ  FUSELAGE DATA
C
      IF (NFU.EQ.0) GO TO 15
      READ (12) NCWB1,(SNBP(I),CSBP(I),THTBP(I),I=1,NRB)
C             READ  ARRAY INFORMATION FOR FOR LINE-SOURCE METHOD
      IF (NFU.NE.1) GO TO 10
      READ(12) FRMAX,FLTHC,NFPOLY,(FXEND(I),I=1,NFPOLY),
     1    ((FCOEF(I,J),J=1,7),I=1,NFPOLY)
      READ  (12) NFSOR,(FXL(I),FSS(I),FDS(I),FDRDX(I),I=1,NFSOR)
     1 ,FBASE,FSUMK,FSUMKD,FXSHLD
      READ  (12) NFSHK,(FXSHK(I),FRSHK(I),I=1,NFSHK)
      GO TO 15
1C    IF (NFU.NE.2) GO TO 15
C             READ  ARRAY INFORMATION FOR SOURCE PANEL METHOD
C             SAVE COPY OF FUSELAGE DATA ON TAPE11.
      CALL FRSTRT (12,2,LASTA)
      CALL FRSTRT (11,1,LASTA)
C
C     READ  PYLON DATA
C
15    IF (NPY.EQ.0) GO TO 20
      READ (12) YPL,CENTER,NCP1,NCPS1
C
C     READ RACK DATA
C
      IF (NRACK.EQ.0) GO TO 20
      READ (12) RLTHC,RRMAX,RIBCR,CRIBCR,SRIBCR,YBRO,XBRO,ZBRO,NRSOR,
     1 NRSHK,RBASE,RSUMK,RSUMKD,RXSHLD
      READ (12) (RXL(I),RSS(I),RDS(I),RDRDX(I),I=1,NRSOR)
      READ (12) (RXSHK(I),RRSHK(I),I=1,NRSHK)
C
C     READ  STORE DATA
C
   20 IF (NSTRS.EQ.0) RETURN
      READ  (12) (SLTHC(I),SRMAX(I),SIBCR(I),CSIBCR(I),SSIBCR(I),
     1    YBSO(I),XBSO(I),ZBSO(I),NUMSTR(I),NSSOR(I),NSSHK(I),NSHAPE(I),
     2    SBASE(I),SSUMK(I),SSUMKD(I),SXSHLD(I),SPHIRR(I),I=1,NSTRS)
      READ  (12) NSHPT,(MSHAPE(I),I=1,NSHPT)
      DO 30 J=1,NSTRS
      IF (MSHAPE(J).GT.50) GO TO 30
C
C             SAVE STORE INFORMATION FOR LINE SOURCE METHOD
      MSOR=NSSOR(J)
      READ  (12) (SXL(I,J),SSS(I,J),SDS(I,J),SDRDX(I,J),I=1,MSOR)
      MSSHK=NSSHK(J)
      READ  (12) (SXSHK(I,J),SRSHK(I,J),I=1,MSSHK)
30    CONTINUE
C
C             SAVE STORE INFORMATION FOR SOURCE PANEL METHOD
      NESHPT = 0
      DO 40 ISHP=1,NSHPT
      IF (MSHAPE(ISHP).LE.50) GO TO 40
      NESHPT = NESHPT+1
      CALL FRSTRT (12,4,LASTA)
      CALL FRSTRT (10,3,LASTA)
40    CONTINUE
      RETURN
      END


      SUBROUTINE REFSHK (NS,SX,SR)
C
C     THIS SUBROUTINE CALCULATES THE LOC  ION AT WHICH THE STORE NOSE
C          SHOCK WHICH IS REFLECTED BY THE WING STRIKES THE STORE AND
C          THE VALUE OF BETA TO USE FOR THE IMAGE STORE NOSE
C
      DIMENSION SX(1),SR(1)
C
      COMMON /CONSTS/ PI,DTOR
```

Figure C-1 (mm)

```
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *    ,CCC,XMOM,ESTMMX,DX,DELX,VSTORE,ESTLGC
      COMMON/GRIDTH/XPTS(1000),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
     1      SWPS(1000),THTNET(1000),DZDX(400)
      COMMON/NUINDX/NRW,NRP,NRH,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1      NCW1,NCP1,NCWB1,NCWS,NCWS1,NCPS1
      COMMON /STORIM/ IMSTOR,BTNOSE,XWN,YWN,ZWN
      COMMON/WGEOM/XBWOC,ZBWO,CRW,ZDIHED
C
C     CALCULATE LOCATION OF STORE NOSE IN FUSELAGE COORDINATE SYSTEM
C
      XXX=XMOM
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XBN=VAR(7)+XI
      YBN=VAR(8)+ETA
      ZBN=VAR(9)+ZETA
C
C     LOCATE THIS POINT IN WING COORDINATE SYSTEM
C
      XWN=XBN-XBWOC
      YWN=YBN
      ZWN=ZBN-ZBWO
C
C     CALCULATE STORE SHOCK LOCATION IN WING CHORDAL PLANE
C
      R=ZWN
      DO 1 J=1,NS
      JJ=J
      IF (R.LT.SR(J)) GO TO 2
    1 CONTINUE
      X=SX(NS)+(SX(NS)-SX(NS-1))*(R-SR(NS))/(SR(NS)-SR(NS-1))
      GO TO 3
    2 JM=JJ-1
      X=SX(JM)+(SX(JJ)-SX(JM))*(R-SR(JM))/(SR(JJ)-SR(JM))
    3 CONTINUE
      XWS=XWN-X
      ZWS=0.0
C
C     AT Y=YWN CALCULATE WING LE AND TE X AND WING CHORD
C
      IPNL=0
      YY=YWN
      IF(YY.GT.0.0) YY=-YY
      IF(YY.GT.YPTS(1)) GO TO 10
      DO 11 I=2,NRWS
      JLER=(I-2)*NCWS1+1
      IF(YY.GE.YPTS(I)) GO TO 15
      IF(YPTS(I).NE.YPTS(I-1)) IPNL=IPNL+1
   11 CONTINUE
   10 IMSTOR=0
      RETURN
   15 DELT=YY-YPTS(I-1)
      WLEX=XPTS(JLER)+DELT*SWPS(JLER)
      JTE=JLER+NCWS
      WTEX=XPTS(JTE)+DELT*SWPS(JTE)
      CHRD=WLEX-WTEX
C
C     IF XWS AHEAD OF WING LE OR BEHIND WING TE SET IMSTOR=0 AND RETURN
C
      IF (XWS.LT.WLEX.AND.XWS.GT.WTEX) GO TO 20
      IMSTOR=0
      RETURN
C
C     CALCULATE WING HALF THICKNESS AT LOCATION OF SHOCK INTERSECTION
C          WITH WING CHORDAL PLANE
C
   20 XC=(WLEX-XWS)/CHRD
      FNCWS=NCWS
      IXC=XC*FNCWS+1.0
      SUM=0.0
      JLE=IPNL*NCWS
      DO 21 I=1,IXC
      ISUB=I+JLE
   21 SUM=SUM+DZDX(ISUB)
      HTHICK=SUM*CHRD/FNCWS
C
C     CALCULATE X AT WHICH SHOCK STRIKES WING SURFACE
C
      RSUR=ZWN-HTHICK
      DO 30 J=1,NS
      JJ=J
      IF (RSUR.LT.SR(J)) GO TO 31
   30 CONTINUE
      X=SX(NS)+(SX(NS)-SX(NS-1))*(RSUR-SR(NS))/(SR(NS)-SR(NS-1))
      GO TO 32
   31 JM=JJ-1
      X=SX(JM)+(SX(JJ)-SX(JM))*(RSUR-SR(JM))/(SR(JJ)-SR(JM))
   32 CONTINUE
C
C     DETERMINE DZ/DX OF THICKNESS ENVELOPE AT THIS LOCATIONS
C
      XWS=XWN-X
      ZWS=HTHICK
      XC=(WLEX-XWS)/CHRD
      IXC=XC*FNCWS+1.0
      ISUB=IXC+JLE
      DZDXL=DZDX(ISUB)
C
C     CALCULATE SHOCK ANGLE AT XWS
C
      IF (JJ.EQ.NS) JM=JJ-1
      DRDXS=(SR(JJ)-SR(JM))/(SX(JJ)-SX(JM))
C
C     CALCULATE SHOCK WAVE REFLECTION ANGLE
      THETAR=(ATAN(DRDXS)+2.0*ATAN(DZDXL))/DTOR
C
C     LOCATE STORE C.G. IN WING SYSTEM ASSUMING YWCG=YWN
C
      XWCG=VAR(7)-XBWOC
      ZWCG=VAR(9)-ZBWO
C
C     DETERMINE INTERSECTION OF REFLECTED SHOCK WAVE AND LINE FROM STORE
C          NOSE THROUGH ABOVE POINT
C
      TALF=(ZWCG-ZWN)/(XWN-XWCG)
      ALF=ATAN(TALF)/DTOR
      IF (THETAR.LE.ALF) GO TO 10
      IF (THETAR.LE.89.5.OR.THETAR.GE.90.5) GO TO 40
      XWR=XWS
      GO TO 41
   40 DRDXR=TAN(DTOR*THETAR)
      XWR=(ZWN-ZWS+XWN*TALF-XWS*DRDXR)/(TALF-DRDXR)
   41 ZWR=ZWN+TALF*(XWN-XWR)
C
C     CALCULATE DISTANCE MEASEURED ALONG AXIS TO REFLECTED SHOCK AND
C          SEE IF THIS POINT IS BEHIND BASE OF STORE
C
      ALFR=DTOR*ALF
      XSR=(XWN-XWR)*COS(ALFR)+(ZWR-ZWN)*SIN(ALFR)
      IF (XSR.GE.ESTLGC) GO TO 10
      IMSTOR=1
C
C     CALCULATE VALUE OF BETA TO USE FOR IMAGE STORE NOSE
C
      BTNOSE=XSR/(2.0*ZWN)
      RETURN
      END


      SUBROUTINE RESVEL(XB,YB,ZB,UTU,VTV,WTW)
C
C     SUBROUTINE TO CALCULATE RESULTANT VELOCITIES AT POINT (XB,YB,ZB,
C     DUE TO ALL AIRCRAFT COMPONENTS EXCEPT SEPARATED STORE
C     COORDINATES XB,YB,ZB AND RETURNED VELOCITIES,UTU,VTV,WTW,
C     ARE IN FUSELAGE SYSTEM
```

Figure C-1(nn)

```
      LOGICAL SUBSON,SUPERS
      LOGICAL INLET
      COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7),
     1 SSUMK(7),SSUMKD(7)
      COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI
     *   ,XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)
      COMMON /BINSHK/ NIS,XISHLD,EALPI,XCLOSD,MAXSHI,NINL(8),PHINL(8)
     *   ,YINL(8),XINL(80),RINL(80)
      COMMON /BODCOM/ AMACH,TAND,CA,XCOR(4),YCOR(4),ZCOR(4)
     1   ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,JP
      COMMON /CONFIG/NFU,NPY,NSTRS,NRACK
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITM,IDEL,NTAP7
     *   ,IAR,IAM,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYH
     *   ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IAO,IDO,ISKO,IYIM,IZIM,ISVN,ISKP
     *   ,NRING,IROW(50)
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /FSOR  /FXL(101),FSS(100),FDS(100),NFSOR
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NHSEGO,NROLL
      COMMON /LOCSET/ FBETAL,RBETAL,SBETAL(7)
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1  ,IDEJ,IDF,ID2
      COMMON /NUFLOW/ FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4
      COMMON /RKGEOM/ RLTHC,RRMAX,RIBCR,CRIBCH,SRIBCR,YRHO,XBRO,ZBRO
      COMMON /RSOR/ NRSOR,RXL(101),RSS(100),RDS(100)
      COMMON /SSOR/ SXL(101,7),SSS(100,7),SDS(100,7),NSSOR(7)
      COMMON /STGEOM/ SLTHC(7),SRMAX(7),SIBCR(7),CSIBCR(7),SSIBCR(7),
     1               YASO(7),XBSO(7),ZBSO(7),NUMSTR(7),NSHAPE(7)
     3               ,NSHPT,MSHAPE(7),SPHIRR(7)
      COMMON /VELARG/ X,Y,Z,U,V,W,EM,TLANC,PYPNL,UP,VP,WP
      COMMON /WGEOM / XBWOC,ZBWO,CRW,ZDTMED
      COMMON          A(12000)
      DIMENSION IA(1),DCS(9,7)
      LOGICAL LA(1),SKIPDC
      EQUIVALENCE (IA(1),A(1),LA(1))
      DATA SKIPDC/.FALSE./
C
C
      UTU=0.0
      VTV=0.0
      WTW=0.0
      BETAO = BETA
      BETAL = BETA
C
C     CALCULATE FUSELAGE SOURCE-DOUBLET VELOCITIES ---------------------
C
      IF (NFU.EQ.0) GO TO 175
      IF (NFU.NE.1) GO TO 120
C             USE LINE SOURCE METHOD FOR VELOCITY CALCULATION
      CALL VELCAL(FSS,FDS,FXL,NFSOR,XB,YB,ZB,UP,VP,WP,FBETAL,
     1   FBASE,FSUMK,FSUMKD)
      GO TO 170
120   IF (NFU.NE.2) GO TO 175
C
C             USE SOURCE PANELING METHOD FOR VELOCITY CALCULATION
      SUBSON=.FALSE.
C             FOR FIELD VELOCITY CALCULATIONS, INCLUDE BOTH INLET AND
C             ADJACENT SET OF PANELS
      NTINL = NINLET+NINBLK
      INLET = LA(IOF+150)
      IF (INLET) GO TO 140
130   BETAO = FBETAL
      CALL FLDVEL (-XB,YB,-ZB,SVN,LSKP,UP,VP,WP,1,IA(IOF+1),1)
      GO TO 160
C
C  IF INLETS EXIST, CHECK TO SEE IF POINT LIES WITHIN INFLUENCE
C             XIS .LE. -XB .LE. XISHLD
C  COMPUTE BETA ASSOCIATED WITH SHOCK PROPAGATING FROM INLET
C
140   CONTINUE
      CALL INLBET (BETAI,XIS,RIS,YB,-ZB)
      DXINL = XINLTE-XINLT
      XISHLD = RIS*BETA+DXINL
      XBTINL = RIS*BTINLT
      XBP = -XB-XINLT
      DX = XISHLD-XBP
      DXS = XISHLD-XIS
      XCLSD = XCLOSD-XINLT
      RCLOSD = XCLSD/BTINLT
      IF (XIS.GE.XBTINL) GO TO 150
C
C  REGION I%   XIS.LT.XATINL
C             IF POINT IS NOT IN REGION OF INFLUENCE OF INLET
C             SHOCK, GO BACK AND USE PROPERTIES DEFINED BY NOSE SHOCK
      IF (XIS.GT.XBP .OR. XBP.GT.XISHLD) GO TO 130
C             FOR POINTS AHEAD OF XBTINL INTERPOLATE FOR  BETA FOR ALL
C             PANELS.  AFT OF XBTINL, USE BETAL FOR ALL PANELS EXCEPT
C             INLET PANELS WHERE BETA=BTINLT
      BETAO = AMAX1(0.01,BETA+((XIS-XCLSD)/(RIS-RCLOSD)-BETA)*DX/DXS)
      XMAP = -XB
      GO TO 155
C
C  REGION II%    XIS.GE.XBTINL
C             CHECK IF POINT IS NOT IN REGION
150   IF (XBTINL.GT.XBP .OR. XBP.GT.XISHLD) GO TO [illegible]
C             MAP REGION FROM XBTINL TO XISHLD FORWARD SUCH THAT XIS
C             MOVES TO XBTINL FOR FIRST INITIAL INFLUENCE.  TRAILING
C             POINT XISHLD REMAINS STATIONARY.
      DXBT = XISHLD-XBTINL
      XMAP = XBP
      IF (DXS.NE.0.0) XMAP=XISHLD-DX*DXBT/DXS
      XMAP = XMAP+XINLT
C             IF POINT AHEAD OF XIS, USE BTINLT
C             FOR POINTS AFT, INTERPOLATE FOR BETAL.  USE BETAL FOR
C             ALL PANELS, EXCEPT INLET PANELS WHICH USE BETA=BTINLT
      IF (XRP.LE.XIS) BETAO=BTINLT
      IF (XRP.GT.XIS)
     *   BETAO = AMAX1(0.01,BETA+((XIS-XCLSD)/(RIS-RCLOSD)-BETA)*DX/DXS)
155   CONTINUE
[illegible] CALL FLDVEL (XMAP,YB,-ZB,SVN,LSKP,UP,VP,WP,1,IA(IDF+1),1)
160   UP = -UP
      WP = -WP
170   CONTINUE
      UTU=UTU+UP
      VTV=VTV+VP
      WTW=WTW+WP
C
C     ------------------------------------------------------        ----
C     TRANSLATE COORDINATES TO WING SYSTEM
C
      XW=XB-XBWOC
      YW=YB
      ZW=ZB-ZBWO
C
C     CALCULATE FUSELAGE INTERFERENCE PANEL VELOCITIES
C
      INUMCH=1
      CALL VELBD2(XW,YW,ZW)
      UTU=UTU+UP
      VTV=VTV+VP
      WTW=WTW+WP
C
C     CALCULATE WING PANEL AND WING THICKNESS VELOCITIES
C
  175 CONTINUE
      INUMCH=1
      CALL VELWP2(XW,YW,ZW)
      UTU=UTU+UP
      VTV=VTV+VP
      WTW=WTW+WP
      CALL VELWT2(XW,YW,ZW)
      UTU=UTU+UP
      VTV=VTV+VP
      WTW=WTW+WP
C
C     CALCULATE PYLON PANEL AND PYLON THICKNESS VELOCITIES ------------
C
```

Figure C-1(oo)

```
      IF (NPY.EQ.0) GO TO 180
      INUMCH=0
      CALL VELPP2(XW,YW,ZW)
      UTU=UTU+UP
      VTV=VTV+VP
      WTW=WTW+WP
      CALL VELPT2(XW,YW,ZW)
      UTU=UTU+UP
      VTV=VTV+VP
      WTW=WTW+WP
C
C     CALCULATE RACK SOURCE-DOUBLET VELOCITIES
C
  180 IF (NRACK.EQ.0) GO TO 200
      TA=XB-XBRO
      TB=ZB-ZBRO
      XR=TA*CRIBCR-TB*SRIBCR
      YR=YB-YBRO
      ZR=TA*SRIBCR+TB*CRIBCR
      CALL VELCAL(RSS,RDS,RXL,NRSOR,XR,YR,ZR,UP,VP,WP,RBETAL,
     1 RBASE,RSUMK,RSUMKD)
      UTU=UTU+UP*CRIBCR+WP*SRIBCR
      VTV=VTV+VP
      WTW=WTW+WP*CRIBCR-UP*SRIBCR
C
C     CALCULATE STORE SOURCE-DOUBLET VELOCITIES -----------------------
C
  200 IF (NSTRS.EQ.0) RETURN
      ISHP1 = 0
      ISHP2 = 0
      DO 260 J=1,NSTRS
      IF (J.EQ.NEJSTR .AND. NSHAPE(J).GT.50) ISHP1=ISHP1+1
      IF (J.EQ.NEJSTR) GO TO 260
C
C  GENERATE DIRECTION COSINES FOR ROLLED AND PITCHED FIXED STORES
C
      IF (SKIPDC) GO TO 210
      SSPHI = SIN(SPHIRR(J))
      CSPHI = COS(SPHIRR(J))
      DCS(1,J) = CSIBCR(J)
      DCS(2,J) = SSPHI*SSIBCR(J)
      DCS(3,J) = -CSPHI*SSIBCR(J)
      DCS(4,J) = 0.0
      DCS(5,J) = CSPHI
      DCS(6,J) = SSPHI
      DCS(7,J) = SSIBCR(J)
      DCS(8,J) = -SSPHI*CSIBCR(J)
      DCS(9,J) = CSPHI*CSIBCR(J)
  210 CALL INTOST (XB-XBSO(J),YB-YBSO(J),ZB-ZBSO(J),XS,YS,ZS,DCS(1,J))
      IF (NSHAPE(J).GT.50) GO TO 220
C
C  USE STORE LINE SOURCE AND DOUBLET ANALYSIS
C
      CALL VELCAL(SSS(1,J),SDS(1,J),SXL(1,J),NSSOR(J),XS,YS,ZS,UP,VP,WP,
     1 SBETAL(J),SBASE(J),SSUMK(J),SSUMKD(J))
      GO TO 255
C
C  USE STORE SOURCE PANEL ANALYSIS
C
  220 IF (NSHAPE(J).NE.NSHAPE(NEJSTR)) GO TO 240
C              USE EJECTED STORE SHAPE
      ISHP1 = ISHP1+1
      BETAL = SBETAL(J)
      CALL FLDVEL (-XS,YS,-ZS,SVN,LSKP,UP,VP,WP,1,NX,ISHP1)
      GO TO 250
C              USE SECOND STORE SHAPE
  240 ISHP2 = ISHP2+1
      BETAL = SBETAL(J)
      CALL FLDVEL (-XS,YS,-ZS,SVN,LSKP,UP,VP,WP,1,IA(ID2+1),ISHP2)
  250 UP = -UP
      WP = -WP
  255 CALL STTOIN (UP,VP,WP,UPS,VPS,WPS,DCS(1,J))
      UTU = UTU+UPS
      VTV = VTV+VPS
      WTW = WTW+WPS
  260 CONTINUE
      SKIPDC = .TRUE.
      RETURN
      END
```

```
      SUBROUTINE ROTBW (YI,ZI,YO,ZO,I)
C
C     THIS SUBROUTINE TRANSFORMS VELOCITIES FROM WING REFERENCE
C     COORDINATE SYSTEM TO INTERFERENCE PANEL COORDINATE SYSTEM AND
C     VICE VERSA.
C
      COMMON /ONE    / DFLTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRR(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLE(250),ZRB(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFRIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
     7   MSWU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
C
C     VELOCITY TRANSFORMS   BIP TO WING
C
      YO=YI*CST2(I)+ZI*SNT2(I)
      ZO=ZI*CST2(I)-YI*SNT2(I)
      RETURN
      END
```

```
      SUBROUTINE ROTFW (YIN,ZIN,YOUT,ZOUT,PHIF)
C
C  THIS SUBROUTINE ROTATES LOCAL VELOCITIES RELATIVE TO CONSTANT
C  U-VELOCITY PANEL CORNERS AND PARALLEL TO WING COORDINATE
C  SYSTEM TO LOCAL COORDINATES IN THE PLANE OF THE PANEL ITSELF AND
C  VICE VERSA.
C
      COMMON /ONE    / DFLTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRR(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLE(250),ZRB(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFRIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
     7   MSWU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
      DATA PIO2/1.570796326795/
C
C  TRANSFORMATION FROM FIN TO WING VELOCITIES
C
      IF(ABS(PHIF-PIO2).LT.TLRNC) GOTO 102
      GOTO 103
  102 COSPHI=0.0
      SINPHI=1.0
      GOTO 105
  103 CONTINUE
      COSPHI=COS(PHIF)
      SINPHI=SIN(PHIF)
  105 CONTINUE
      YOUT=YIN*COSPHI-ZIN*SINPHI
      ZOUT=ZIN*COSPHI+YIN*SINPHI
      RETURN
      END
```

```
      SUBROUTINE ROTWB(YI,ZI,YO,ZO,I)
C
```

Figure C-1(pp)

```
C     THIS SUBROUTINE TRANSFORMS VELOCITIES FROM WING REFERENCE
C     COORDINATE SYSTEM TO BODY INTERFERENCE PANEL COORDINATE SYSTEM
C
      COMMON /ONE   / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRR(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLR(250),ZRR(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFRIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
     7   MSWU,MSWD,NBIP,NCW,NMP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
C
C     VELOCITY TRANSFORMATION FROM  WING TO BIP
C
      YO = YI*CST2(I)-ZI*SNT2(I)
      ZO = ZI*CST2(I)+YI*SNT2(I)
      RETURN
      END


      SUBROUTINE ROTWF (YIN,ZIN,YOUT,ZOUT,PHIF)
C
C  THIS SUBROUTINE ROTATES LOCAL VELOCITIES RELATIVE TO CONSTANT
C  U-VELOCITY PANEL CORNERS AND PARALLEL TO WING COORDINATE
C  SYSTEM TO LOCAL COORDINATES IN THE PLANE OF THE PANEL ITSELF AND
C  VICE VERSA.
C
      COMMON /ONE   / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRB(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLR(250),ZRR(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLR,U,V,W,UCHR,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
     7   MSWU,MSWD,NBIP,NCW,NMP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
      DATA PIO2/1.570796326795/
C
C  TRANSFORMATION FROM WING TO FIN VELOCITIES
C  WING TO FIN
C
      IF(ABS(PHIF-PIO2).LT.TLRNC) GOTO 100
      GOTO 101
  100 COSPHI=0.0
      SINPHI=1.0
      GOTO 104
  101 CONTINUE
      COSPHI=COS(PHIF)
      SINPHI=SIN(PHIF)
  104 CONTINUE
      YOUT=YIN*COSPHI+ZIN*SINPHI
      ZOUT=ZIN*COSPHI-YIN*SINPHI
      RETURN
      END


      SUBROUTINE SDSTN2 (UT,VT,WT,LSKP,IMAGE)
C
C  ROUTINE TO ORGANIZE SOLUTION FOR PANEL STRENGTHS AND  FORCES
C  FOR ELLIPTIC STORE
C
      LOGICAL LSKP(1),IMAGE,IMAGVN
      DIMENSION UT(48),VT(48),WT(48)
      COMMON /BFORCE/ CNSA,CYSA,CLMSB,CLNSB,CLLSB
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *   ,COC,XMOM,ESTRMX,DX,DELX,VSTORE,ESTLGC
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NADIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *   ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IAO,IDO,ISKO,IYIM,IZIM,ISVN,ISKP
     *   ,NRING,IROW(50)
       COMMON /FORM  / CN,CT,CM,CYB,CLROLL,CNYAW,CL,CD,DXN
       COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NMSEGO,NROLL
       COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
      1  ,IDEJ,IDF,ID2
       COMMON             A(12000)
C
C  COMPUTE NORMAL VELOCITIES AT CONTROL POINTS FOR  BOUNDARY CONDITION.
C
       JVB = IVB-1
       JSVN = ISVN-1
       DO 100 J=1,NBODY
       JSVN = JSVN+1
       A(JSVN) = 0.0
       LSKP(J) = .FALSE.
       JVB = JVB+1
 100   A(JVB) = -VNORM (J,UT,VT,WT,A(ITH),A(IDEL))
C
C  SOLVE MATRIX FOR PANEL STRENGTHS
C
       IG = IGB(NEJGB)
       IMAGVN = IMAGE
       REWIND 8
       REWIND 9
       CALL SMARCH (A(IG),A(IVB),IAN,IMAGVN)
C
C  SOLVE FOR U,V,W VELOCITIES AT CONTROL POINTS
C
       CALL SOLVUV (A(IU),A(IV),A(IW),A(IAN),A(IG),NBODY,.TRUE.)
C
C  ADD CONTRIBUTION OF DAMPING TERMS TO PERTURBATION VELOCITIES
C
       IF (NDAMP.NE.0) CALL VELDMP (A(IU),A(IV),A(IW)
      *    ,A(IXPT),A(IYPT),A(IZPT),NBODY,AMOM,VSTORE,VAR)
C
C  COMPUTE PRESSURES ON SURFACE
C
       CALL PRESS (NBODY,A(IU),A(IV),A(IW),A(ICP),CPSTAG,CPCRIT,CPVAC)
C
C  CALCULATE FORCES AND MOMENTS
C
       IXC1 = IXC(1)
       CALL FORMOM (A(IXPT),A(IYPT),A(IZPT),A(ITH),A(IDEL),A(IAR)
      *    ,A(ICP),A(IXC1))
C
C  DEFINE FORCE COEFFICIENTS
C
       CNSA  = CN
       CYSB  = CYB
       CLMSB = CM
       CLNSB = CNYAW
       CLLSB = 0.0
       IF (NROLL.NE.0) CLLSB=CLROLL
       RETURN
       END


       SUBROUTINE SDSTRN(XSMOL)
C
C      THIS SUBROUTINE CALCULATES THE SOURCE AND DOUBLET STRENGTHS
C           FOR THE SEPARATED STORE INCLUDING THE EFFECT OF THE
C           IMAGE STORE
C
       LOGICAL IMAGE
       COMMON /CFORCE/ CNX(80),CYX(80),XCP(81),DRDXCP(81),RCP(81)
      *    ,UT(81),VT(81),WT(81)
       COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
      *    ,COC,XMOM,ESTRMX,DX,DELX,VSTORE,ESTLGC
       COMMON /NEWFOR/ NTHETA,DTHETA,THETA(13?),STHETA(13?),CTHETA(13?),
      1REFL,SREF,NCONPT
```

Figure C-1(qq)

```
      COMMON /SORDUB/ BETAA,SRS(150),VRATSA,BETAS,BSS,DSW(150),
     1 DSV(150),XS(150),XD(150),BETAA1,BETAS1,RSS1
      COMMON /STORIC/ BTNOSC,RSSC
      COMMON/STORIF/ IMFSTR,BTNOSF,RSS,YBN,ZBN,SFAC
      COMMON /STOWIM/ IMSTOR,BTNOSE,XWN,YWN,ZWN,IMAGE
      COMMON /STORSD/ NXBODY,XBD(151),RBD(151),NPOLY,XEND(7),COEF(7,7),
     1 XBC(150),RBC(150),DRDXBC(150),NSORCE,DRDXBD(151)
      COMMON /STRBET/ BTAL,BTSL
      COMMON /STRBTC/ BTALC,BTSLC
      COMMON /STRBTF/ BTALF,BTSLF
      COMMON /TRMSSD/ ARG,ROOT,ACOSH
C
C     LOCATE WING IMAGE STORE IN STORE COORDINATE SYSTEM
C
      FMACHA=SQRT(BETAA*BETAA+1.0)
      FMACHS=SQRT(BETAS*BETAS+1.0)
      IF (IMSTOR.EQ.0) GO TO 1
      RFP=ZWN-ZBN
      XSHWI=RFP*BTNOSE
      XINAI=RFP*BETAA+XSHOL
      XINSI=RFP*BETAS+XSHOL
      FMSHI=SQRT(BTNOSE*BTNOSE+1.0)
      STHET=SIN(-VAR(12))
      CTHET=COS(VAR(12))
C
C     LOCATE FUSELAGE IMAGE STORE IN STORE COORDINATE SYSTEM
C
    1 IF (IMFSTR.EQ.0) GO TO 2
      RFPF=RSS
      RFPC=RSSC
      XSHWIF=RFPF*BTNOSF
      XINAIF=RFPF*BETAA+XSHOL
      XINSIF=RFPF*BETAS+XSHOL
      FMSHIF=SQRT(BTNOSF*BTNOSF+1.0)
      THET=-VAR(12)+ATAN(-YBN/ZBN)
      STHETF=SIN(THET)
      CTHETF=COS(THET)
    2 CONTINUE
C
C     CALCULATE FIRST SOURCE AND DOUBLET STRENGTHS
C
      DR=DRDXBC(1)
      XOR=XBC(1)/RBC(1)
      ARG=XOR/BETAA
      CALL SDTRMS
      SRS(1)=UT(1)* VRATSA*DR/(BETAA  *ROOT+ACOSH*DR)
      ARG=XOR/BETAS
      CALL SDTRMS
      TA=0.5*BSS*(ACOSH-ARG*ROOT)+DR*BETAS*ROOT
      DSW(1)=WT(1)/TA
      DSV(1)=VT(1)/TA
C
C     CALCULATE REMAINDER OF SOURCE AND DOUBLET STRENGTHS
C
      DO 30 N=2,NCONPT
      DR=DRDXBC(N)
      XX=XBC(N)
      BAR=BETAA*RBC(N)
      BSR=BETAS*RBC(N)
      SUMS=0.0
      SUMDW=0.0
      SUMDV=0.0
      NM=N-1
      DO 31 J=1,NM
      ARG=(XX-XS(J))/BAR
      CALL SDTRMS
      SUMS=SUMS+SRS(J)*(BETAA*ROOT+ACOSH*DR)
      ARG=(XX-XD(J))/BSR
      CALL SDTRMS
      TA=0.5*BSS*(ACOSH+ARG*ROOT)+DR*BETAS*ROOT
      SUMDW=SUMDW+DSW(J)*TA
   31 SUMDV=SUMDV+DSV(J)*TA
C
C     ADD IN IMAGE STORE VELOCITIES
C
      XFP=XX
C
C     WING IMAGE STORE
C
      IF (IMSTOR.EQ.0) GO TO 10
      BTAL=BETAA
      BTSL=BETAS
      IF (XFP.LT.XSHWI) GO TO 11
      IF (XFP.GT.XSHWI) GO TO 12
      BTAL=BTNOSE
      BTSL=BTNOSE
      GO TO 11
   12 IF (XFP.GE.XINSI) GO TO 11
      FMLS=FMSHI+(FMACHS-FMSHI)*(XFP-XSHWI)/(XINSI-XSHWI)
      BTSL=SQRT(FMLS*FMLS-1.0)
      IF (XFP.GE.XINAI) GO TO 11
      FMLA=FMSHI+(FMACHA-FMSHI)*(XFP-XSHWI)/(XINAI-XSHWI)
      BTAL=SQRT(FMLA*FMLA-1.0)
   11 CONTINUE
      FAC=1.0
      CALL IMSVEL (NM,XFP,RFP,CTHET,STHET,BTAL,BTSL,USD,VSD,WSD,FAC)
      UT(N)=UT(N)+USD
      VT(N)=VT(N)+VSD
      WT(N)=WT(N)+WSD
C
C     FUSELAGE IMAGE STORE
C
   10 IF (IMFSTR.EQ.0) GO TO 50
      BTALF=BETAA
      BTSLF=BETAS
      IF (XFP.LT.XSHWIF) GO TO 13
      IF (XFP.GT.XSHWIF) GO TO 14
      BTALF=BTNOSF
      BTSLF=BTNOSF
      GO TO 13
   14 IF (XFP.GE.XINSIF) GO TO 13
      FMLS=FMSHIF+(FMACHS-FMSHIF)*(XFP-XSHWIF)/(XINSIF-XSHWIF)
      BTSLF=SQRT(FMLS*FMLS-1.0)
      IF (XFP.GE.XINAIF) GO TO 13
      FMLA=FMSHIF+(FMACHA-FMSHIF)*(XFP-XSHWIF)/(XINAIF-XSHWIF)
      BTALF=SQRT(FMLA*FMLA-1.0)
   13 BTALC=BTALF*RSS/RSSC
      BTSLC=BTSLF*RSS/RSSC
      CALL IMSVEL (NM,XFP,RFPF,CTHETF,STHETF,BTALF,BTSLF,USD,VSD,WSD,
     1 SFAC)
      UT(N)=UT(N)+USD
      VT(N)=VT(N)+VSD
      WT(N)=WT(N)+WSD
      FAC=0.0
      CALL IMSVEL (NM,XFP,RFPC,CTHETF,STHETF,BTALC,BTSLC,USD,VSD,WSD,
     1 FAC)
      UT(N)=UT(N)-USD
      VT(N)=VT(N)-VSD
      WT(N)=WT(N)-WSD
C
C     CALCULATE STRENGTHS
C
   50 ARG=(XX-XS(N))/BAR
      CALL SDTRMS
      SRS(N)=(UT(N)*VRATSA*DR-SUMS)/(BETAA*ROOT+ACOSH*DR)
      ARG=(XX-XD(N))/BSR
      CALL SDTRMS
      TA=0.5*BSS*(ACOSH+ARG*ROOT)+DR*BETAS*ROOT
      DSW(N)=(WT(N)-SUMDW)/TA
      DSV(N)=(VT(N)-SUMDV)/TA
   30 CONTINUE
      RETURN
      END

      SUBROUTINE SDTRMS                                                 SDTR  10
```

Figure C-1(rr)

```
C
C     THIS SUBROUTINE EVALUATES THE SQUARE ROOT AND INVERS COSH TERMS
C          APPEARING IN THE SOURCE AND DOUBLET EXPRESSIONS
C
      COMMON /TRNSSD/ ARG,ROOT,ACOSH
C
      IF (ARG.LE.1.0) GO TO 1
      ROOT=SQRT(ARG*ARG-1.0)
      ACOSH=ALOG(ARG+ROOT)
      RETURN
    1 ROOT=0.0
      ACOSH=0.0
      RETURN
      END



      SUBROUTINE SEMFOR
C     SUBROUTINE TO CALCULATE EMPENNAGE FORCES
C
      DIMENSION UTL(11,4),VTL(11,4),WTL(11,4),VN(11,4),SARG(11)
      COMMON /BVELFS/ VX,VY,VZ,VRATS,RVX,RVY,RVZ
      COMMON /CONSTS/ PI,DTOR
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *   ,CDC,XMOM,ESTRMX,DX,DELX,VSTORE,ESTLGC
      COMMON /EFORC1/ CYEM,CNEM,CLLEM,CLMEM,CLNEM
      COMMON /LETEM/FMXT3,FMXT4
      COMMON /EMPCON/ RFIN(11),YTAIL(11,4),ZTAIL(11,4),FROLE(4),FCONA,
     1FCONB,FCONC,CCLY(11),CCLS(11),XTAIL1,DYFIN
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,PHIROL,MSF,IPLNR,CLALPH
      COMMON /FLOW/ALFACH,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NMSEG,NMSEGO,NROLL
      COMMON/NUFLOW/FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4
C
      DO 1 J=1,4
      DO 1 K=1,MSF
      UTL(K,J)=0.0
      VTL(K,J)=0.0
    1 WTL(K,J)=0.0
      XXX=XTAIL+XMOM
      JMAX=4
      IF (IPLNR.EQ.1) JMAX=2
C      CALCULATE THE BETAS TO BE USED IN THE FIN CONTROL POINT VELOCITY
C     CALCULATIONS
C
      CALL STTOIN(XXX,.0,.0,XI,ETA,ZETA,DC)
      XB=VAR(7)+XI
      YB=VAR(8)+ETA
      ZB=VAR(9)+ZETA
      CALL BVARIA(XB,YB,ZB)
C
C     CALCULATE LOCAL WING THICKNESS MACH NUMBER AT FIN XB
C
      IF(XB.GT.XT3.OR.XB.LT.XT4) GOTO 26
      FNUMCH=FMXT3+(FMXT4-FMXT3)*(XB-XT3)/(XT4-XT3)
      BTSQNU=FNUMCH*FNUMCH-1.
      BETANU=SQRT(BTSQNU)
      GOTO 27
   26 FNUMCH=FMACH
      BETANU=BETA
      BTSQNU=BETASQ
   27 CONTINUE
C
C     CALCULATE VELOCITIES IN STORE COORDINATE SYSTEM
C
      ISMTH=0
      DO 2 J=1,JMAX
      DO 2 K=1,MSF
      YYY=YTAIL(K,J)
      ZZZ=ZTAIL(K,J)
      CALL STTOIN (XXX,YYY,ZZZ,XI,ETA,ZETA,DC)
      XB=VAR(7)+XI
      YB=VAR(8)+ETA
      ZB=VAR(9)-ZETA
C
C     CALCULATE PERTURBATION VELOCITY FIELD
C
      CALL RESVEL(XB,YB,ZB,URES,VRES,WRES)
      CALL INTOST(URES,VRES,WRES,UTL(K,J),VTL(K,J),WTL(K,J),DC)
      UTL(K,J)=VRATS*UTL(K,J)
      VTL(K,J)=VRATS*VTL(K,J)
      WTL(K,J)=VRATS*WTL(K,J)
      CALL ZIMAGE(XXX,YYY,ZZZ,UIM,VIM,WIM)
      UTL(K,J)=UTL(K,J)-UIM
      VTL(K,J)=VTL(K,J)+VIM
      WTL(K,J)=WTL(K,J)-WIM
    2 CONTINUE
C
C     ADD FREE-STREAM COMPONENTS TO PERTURBATION VELOCITIES
C     RESULTING VELOCITIES ARE POSITIVE IN -XB, +YB, AND -ZB DIRECTIONS
C
      DO 4 J=1,JMAX
      DO 4 K=1,MSF
      UTL(K,J)=-UTL(K,J)+RVX
      VTL(K,J)=VTL(K,J)-RVY
    4 WTL(K,J)=-WTL(K,J)+RVZ
C
C     ADD IN Q AND R DAMPING
C
      IF (NDAMP.EQ.0) GO TO 6
      DUM=-XXX/VSTORE
      FA=VAR(6)*DUM
      FB=VAR(5)*DUM
      DO 5 J=1,JMAX
      DO 5 K=1,MSF
      VTL(K,J)=VTL(K,J)+FA
    5 WTL(K,J)=WTL(K,J)+FB
C
C     DETERMINE VELOCITY COMPONENTS NORMAL TO PANELS EXCLUDING
C          ROLL DAMPING CONTRIBUTION
C
    6 DUM=-1.0
      DO 7 J=1,JMAX
      SPHI=SIN(FROLE(J))
      CPHI=COS(FROLE(J))
      DUM=-DUM
      DO 7 K=1,MSF
    7 VN(K,J)=DUM*(WTL(K,J)*SPHI-VTL(K,J)*CPHI)
C
C     CALCULATE VELOCITIES ON BODY AXIS AT X=XTAIL
C
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XB=VAR(7)+XI
      YB=VAR(8)+ETA
      ZB=VAR(9)+ZETA
      CALL RESVEL (XB,YB,ZB,URES,VRES,WRES)
      CALL INTOST (URES,VRES,WRES,UBA,VBA,WBA,DC)
      VBA=VRATS*VBA
      WBA=VRATS*WBA
      CALL ZIMAGE(XXX,.0,.0,UIM,VIM,WIM)
      VBA=VBA+VIM
      WBA=WBA-WIM
      VSO=VBA-RVY
      WSO=-WBA+RVZ
      IF (NDAMP.EQ.0) GO TO 20
      DUM=-XXX/VSTORE
      VSO=VSO+VAR(6)*DUM
      WSO=WSO+VAR(5)*DUM
   20 CONTINUE
C
C     CALCULATE EMPENNAGE NORMAL FORCE, SIDE FORCE, PITCHING MOMENT,
C          AND YAWING MOMENT
C
      ANG=DTOR*PHIROL
      WO=WSO*COS(ANG)+VSO*SIN(ANG)
      VO=-WSO*SIN(ANG)+VSO*COS(ANG)
      A2=RADAV**2
```

Figure C-1(ss)

```
      DO 8 K=1,MSF
      AS=(VN(K,1)+VN(K,2))/2.0+W0*A2/(RFIN(K)**2)
    8 SARG(K)=AS*CCL3(K)
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      CNEM=FCONA*SUM
      CLMEM=-FCONB*SUM*FCONC
      CYEM=0.0
      CLNEM=0.0
      IF (IPLNR.EQ.1) GO TO 10
      DO 9 K=1,MSF
      BS=(VN(K,3)+VN(K,4))/2.0+V0*A2/(RFIN(K)**2)
    9 SARG(K)=BS*CCL3(K)
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      CYEM=FCONA*SUM
      CLNEM=-FCONB*SUM*FCONC
   10 TA=CNEM
      TB=CYEM
      CA=COS(ANG)
      SA=SIN(ANG)
      CNEM=TA*CA-TB*SA
      CYEM=TA*SA+TB*CA
      TA=CLMEM
      TB=CLNEM
      CLMEM=TA*CA-TB*SA
      CLNEM=TA*SA+TB*CA
C
C     CALCULATE ROLLING MOMENT
C
      CLLEM=0.0
      IF (NROLL.EQ.0) RETURN
      DO 11 K=1,MSF
      ABU=(VN(K,1)-VN(K,2))/2.0
      IF (NDAMP.GT.0) ABU=ABU+VAR(4)*RFIN(K)/VSTORE
      IF (IPLNR.EQ.1) GO TO 12
      ABU=ABU+(VN(K,3)-VN(K,4))/2.0
      IF (NDAMP.GT.0) ABU=ABU+VAR(4)*RFIN(K)/VSTORE
   12 SARG(K)=ABU*CCL5(K)
   11 CONTINUE
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      CLLEM=-FCONB*SUM
      RETURN
      END


      SUBROUTINE SEMPIN
C     SUBROUTINE TO INITIALIZE FOR EMPENNAGE CALCULATION
C
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *   ,COC,XMOM,ESTRMX,DX,DELX,VSTORE,ESTLGC
      COMMON /CONSTS/ PI,DTOR
      COMMON /EMPCON/ RFIN(11),YTAIL(11,4),ZTAIL(11,4),FROLE(4),FCONA,
     1FCONB,FCONC,CCL3(11),CCL5(11),XTAILI,DYFIN
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,PHIROL,MSF,IPLNR,CLALPH
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NMSEG,NMSEGO,NROLL
C
C     LOCATE SPANWISE CONTROL POINTS
C
      TA=MSF-1
      TB=FINSS-RADAV
      DYFIN=TB/TA
      DO 1 J=1,MSF
      TC=J-1
    1 RFIN(J)=RADAV+TC*DYFIN
C
C     LOCATE CONTROL POINTS ON FINS IN STORE COORDINATE SYSTEM
C
      IF (XTAIL.GT.0.0) XTAIL=-XTAIL
      FROLE(1)=DTOR*(PHIROL+90.0)
      FROLE(2)=DTOR*(PHIROL+270.0)
      FROLE(3)=DTOR*(PHIROL+180.0)
      FROLE(4)=DTOR*(PHIROL)
      JMAX=4
      IF (IPLNR.EQ.1) JMAX=2
      DO 2 J=1,JMAX
      SPHI=SIN(FROLE(J))
      CPHI=COS(FROLE(J))
      DO 2 K=1,MSF
      YTAIL(K,J)=RFIN(K)*SPHI
    2 ZTAIL(K,J)=-RFIN(K)*CPHI
C
C     COMPUTE CONSTANTS
C
      FCONA=CLALPH/(3.1415927*TB*TB)
      FCONB=FCONA/(2.0*ESTRMX)
      FCONC=-XTAIL-XMOM
      SHS=FINSS**2
      A4=RADAV**4
      A2=RADAV**2
      TOVPI=2.0/3.1415927
      DO 3 J=1,MSF
      R2=RFIN(J)**2
      ARG=(SHS*R2-A4)*(SHS-R2)/(SHS*R2)
    3 CCL3(J)=4.0*SQRT(ARG)
      IF (NROLL.EQ.0) RETURN
      IF (IPLNR.EQ.0) GO TO 10
C
C     COMPUTE LOADING COEFFICIENTS FOR ROLLING MOMENT CALCULATION
C
C     PLANAR EMPENNAGE
C
      TA=1.0+TOVPI*ACOS(2.0*FINSS*RADAV/(SHS+A2))
      TB=(SHS-A2)/(SHS+A2)
      DO 4 J=1,MSF
      R=RFIN(J)
      R2=R*R
      CCL5(J)=TA*(R+A2/R)*CCL3(J)/4.0
      IF (J.EQ.1.OR.J.EQ.MSF) GO TO 4
      TC=(R2+A2)*TB/(R2-A2)
      CCL5(J)=CCL5(J)+TOVPI*((R-A2/R)**2)*ALOG(TC+SQRT(TC*TC-1.0))
    4 CONTINUE
      RETURN
C
C     CRUCIFORM EMPENNAGE
C
   10 CAPRSQ=(SHS+A4/SHS)/4.0
      CTWGAM=A2/(2.0*CAPRSQ)
      TWGAM=ACOS(CTWGAM)
      STWGAM=SIN(TWGAM)
      TTWGAM=STWGAM/CTWGAM
      AK1=STWGAM
      CALL CEL1 (AKK1,STWGAM,IER)
      R2=RFIN(1)**2
      R4=R2*R2
      SH4=SHS*SHS
      TWOTHE=ACOS(SHS*(A4+R4)/(R2*(A4+SH4)))
      CCL5(1)=4.0*TOVPI*CAPRSQ*(-CTWGAM*ALOG(CTWGAM)+0.5*AKK1*SIN(2.0*
     1TWOTHE))
      CK=SQRT(1.0-AK1*AK1)
      CKCK=CK*CK
      CALL CEL2 (EPIO2K,AK1,1.0,CKCK,IER)
      DO 11 J=2,MSF
      R2=RFIN(J)**2
      R4=R2*R2
      CTWTHE=SHS*(A4+R4)/(R2*(A4+SH4))
      TWTHE=ACOS(CTWTHE)
      STWTHE=SIN(TWTHE)
      TTWTHE=TAN(TWTHE)
      ARG1=STWTHE/STWGAM
      ARG2=TTWTHE/TTWGAM
      ARG3=ASIN(ARG1)
      CCL5(J)=4.0*TOVPI*CAPRSQ*(CTWTHE*0.5*ALOG((1.0+ARG1)/(1.0-ARG1))
     1-CTWGAM*0.5*ALOG((1.0+ARG2)/(1.0-ARG2)))
      TANA1=TAN(ARG3)
      CALL ELI1 (FAK,TANA1,CK)
      CALL ELI2 (EAK,TANA1,CK,1.0,CKCK)
      CCL5(J)=CCL5(J)+2.0*TOVPI*CAPRSQ*(AKK1*SIN(2.0*TWTHE)-2.0*STWGAM
```

Figure C-1(tt)

```
     1*COS(ARG3)*(AKK1*EAK-EPIO2K*FAK))
  11 CONTINUE
     RETURN
     END



      SUBROUTINE SFORCE
C
C     SUBROUTINE TO CALCULATE AERODYNAMIC FORCES AND MOMENTS ON A
C          CIRCULAR BODY USING THE THREE-DIMENSIONAL METHOD USING
C          LINE SOURCES AND DOUBLETS
C
      LOGICAL IMAGE
      DIMENSION UP(80),VP(80),WP(80)
      DIMENSION CP(37),FUNC(37)
      DIMENSION SX(50),SR(50)
      COMMON /BFORCE/ CNSB,CYSB,CLMSB,CLNSB,CLLSB
      COMMON /BVELFS/ VX,VY,VZ,VRATS,RVX,RVY,RVZ
      COMMON /CFORCE/ CNX(80),CYX(80),XCP(81),DRDXCP(81),RCP(81)
     *   ,UT(81),VT(81),WT(81)
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *   ,CDC,XMOM,ESTRMX,DX,DELX,VSTORE,ESTLGC
      COMMON /FLOW/ ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NMSEGO,NROLL
      COMMON /LETEM/ FMXT3,FMXT4
      COMMON /NEWFOR/ NTHETA,DTHETA,THETAD(37),STHETA(37),CTHETA(37),
     1REFL,SREF,NCONPT
      COMMON/NUFLOW/FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4
      COMMON /SORDUB/ BETAA,SRS(150),VRATSA,BETAS,BSS,DSW(150),
     1 DSV(150),XS(150),XD(150),BETAAI,BETASI,BSSI
      COMMON /SSHOCK/ NSSHK(7),SXSHK(50,7),SRSHK(50,7),SDRDX(101,7)
      COMMON /STORIC/ BTNOSC,RSSC
      COMMON/STORIF/ IMFSTR,BTNOSF,RSS,YBN,ZBN,SFAC
      COMMON /STORIM/ IMSTOR,BTNOSE,XWN,YWN,ZWN,IMAGE
      COMMON /STORSD/ NXBODY,XBD(151),RBD(151),NPOLY,XEND(7),COEF(7,7),
     1 XBC(150),RBC(150),DRDXBC(150),NSORCE,DRDXBD(151)
      COMMON /STRBET/ BTAL,BTSL
      COMMON /STRBTC/ BTALC,BTSLC
      COMMON /STRATF/ BTALF,BTSLF
      COMMON /TRNSSD/ ARG,ROOT,ACOSH
      COMMON /WGEOM/ XRWOC,ZBWO,CRW,ZDIHED
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
C
  700 FORMAT(/// 5X,10H***** AT BASE OF BODY RADIAL DISTANCE TO MACH CO
     1NE EMANATING FROM BODY NOSE IS LESS THAN MAXIMUM BODY RADIUS/13X,
     223HCHECK BODY INPUT DATA AND MACH NUMBER)
  707 FORMAT(///5X,69H***** MACH CONE-BODY MERIDIAN INTERSECTION NOT FOU
     1ND AFTER 100 TRIALS)
C
C
C     CALCULATE MACH NUMBERS AND BETAS FOR USE IN SOURCE AND DOUBLET
C          STRENGTH DETERMINATION
C
      VRATA=VINF/VX
      FMACHA=FMACH/VRATA
      FMACHS=FMACH/VRATS
      BETAA=SQRT(FMACHA*FMACHA-1.0)
      BETAS=SQRT(FMACHS*FMACHS-1.0)
C
C     CALCULATE SOURCE AND DOUBLET ORIGINS
C
      IF (BETAS*ESTRMX.LT.ESTLGC) GO TO 9
      WRITE (6,700)
      STOP
    9 XS(1)=0.0
      CALL SHAPE(XS(1),NPOLY,XEND,COEF,RR,DRDXBD(1))
      XD(1)=0.0
      DO 5 N=2,NXBODY
      XS(N)=XBD(N)-BETAA  *RBD(N)
      XXX=XS(N)/ESTLGC
      CALL SHAPE(XXX,NPOLY,XEND,COEF,RR,DRDXBD(N))
    5 XD(N)=XBD(N)-BETAS  *RBD(N)
C
C     IF CONTROL POINTS LIE OUTSIDE THE MACH CONE WITH ITS ORIGIN AT THE
C          BODY NOSE, REVISE LAYOUT FOR BODY DEFINITION POINTS,ETC.
C
      NCONPT=NXBODY-1
      IF (BETAS*RBC(1).LT.XBC(1)) GO TO 20
C
C     DETERMINE X-STATION OF INTERSECTION OF BODY NOSE MACH CONE AND
C          BODY SURFACE
C
      DO 80 I=2,NSORCE
      IF ((BETAS*RBC(I)-XBC(I)).LT.0.0) GO TO 81
   80 CONTINUE
   81 IAFT=I
      IBFR=I-1
      DELTX=0.1*(XBC(IAFT)-XBC(IBFR))
      ITRY=0
      XTRY=XBC(IBFR)
   84 RMCNE=XTRY/BETAS
      XOL=XTRY/ESTLGC
      CALL SHAPE(XOL,NPOLY,XEND,COEF,ROL,SLPE)
      ITRY=ITRY+1
      RI=ROL*ESTLGC
      ERR=RI-RMCNE
      IF (ITRY.GE.100) GO TO 82
      IF (ERR.LT.0.0) GO TO 83
      XTRY=XTRY+DELTX
      GO TO 84
   82 WRITE (6,707)
      STOP
   83 IF (ABS(ERR).LT.(0.01*RI)) GO TO 85
      XTRY=XTRY-DELTX
      DELTX=0.1*DELTX
      GO TO 84
   85 XINT=XTRY
C
C     REVISE LAYOUT OF BODY DEFINITION POINTS, CONTROL POINTS, AND
C          ORIGINS OF LINE SINGULARITIES
C
      SBODY=ESTLGC-XINT
      FN=NSORCE
      DEL=SBODY/FN
      XBD(2)=XINT+DEL
      DO 86 I=3,NSORCE
   86 XBD(I)=XBD(I-1)+DEL
      DO 87 I=1,NXBODY
      XOL= XBD(I)/ESTLGC
      CALL SHAPE(XOL,NPOLY,XEND,COEF,ROL,SLPE)
   87 RBD(I)=ROL*ESTLGC
      XBC(1)=0.5*(XINT+XBD(2))
      XOL=XBC(1)/ESTLGC
      CALL SHAPE(XOL,NPOLY,XEND,COEF,ROL,DRDXBC(1))
      RBC(1)=ROL*ESTLGC
      DO 88 I=2,NSORCE
      XBC(I)=0.5*(XBD(I)+XBD(I+1))
      XOL=XBC(I)/ESTLGC
      CALL SHAPE(XOL,NPOLY,XEND,COEF,ROL,DRDXBC(I))
   88 RBC(I)=ROL*ESTLGC
      XS(1)=0.0
      CALL SHAPE(XS(1),NPOLY,XEND,COEF,ROL,DRDXBD(1))
      XD(1)=0.0
      DO 89 I=2,NXBODY
      XS(I)=XBD(I)-BETAA*RBD(I)
      XOL=XS(I)/ESTLGC
      CALL SHAPE(XOL,NPOLY,XEND,COEF,ROL,DRDXBD(I))
   89 XD(I)=XBD(I)-BETAS*RBD(I)
   20 CONTINUE
C
C     CALCULATE PERTURBATION VELOCITY FIELD ON STORE AXIS AT AXIAL
C          LOCATIONS OF CONTROL POINTS
C
      YY=VAR(8)
      ZZ=VAR(9)-ZWN
      CALL NUMACH(YY,ZZ)
```

Figure C-1(uu)

```
      ISMTH=0                                                           SFOR1410
C                                                                       SFOR1420
C     CALCULATE LOCATION AT WHICH THE STORE NOSE SHOCK WAVE WHICH       SFOR1430
C          REFLECTS OFF THE WING STRIKES THE STORE                      SFOR1440
C                                                                       SFOR1450
      SALF=SIN(ALFACR)                                                  SFOR1460
      CALF=COS(ALFACR)                                                  SFOR1470
      NS=NSSHK(NEJSTR)                                                  SFOR1480
      DO 17 N=1,NS                                                      SFOR1490
      X=SXSHK(N,NEJSTR)                                                 SFOR1500
      R=SRSHK(N,NEJSTR)                                                 SFOR1510
      SX(N)=X*CALF-R*SALF                                               SFOR1520
      SR(N)=X*SALF+R*CALF                                               SFOR1530
   17 CONTINUE                                                          SFOR1540
      CALL REFSHK(NS,SX,SR)                                             SFOR1550
C                                                                       SFOR1560
C     CALCULATE THE LOCATION AT WHICH THE STORE NOSE SHOCK WAVE WHICH   SFOR1570
C          REFLECTS OFF THE FUSELAGE STRIKES THE STORE                  SFOR1580
C                                                                       SFOR1590
      ZBN=ZWN+ZBW0                                                      SFOR1600
      RADIUS=SQRT(YWN*YWN+ZBN*ZBN)                                      SFOR1610
      CPHI=ZBN/RADIUS                                                   SFOR1620
      ALFA=CPHI*ALFACR                                                  SFOR1630
      SALF=SIN(ALFA)                                                    SFOR1640
      CALF=COS(ALFA)                                                    SFOR1650
      DO 18 N=1,NS                                                      SFOR1660
      X=SXSHK(N,NEJSTR)                                                 SFOR1670
      R=SRSHK(N,NEJSTR)                                                 SFOR1680
      SX(N)=X*CALF-R*SALF                                               SFOR1690
   18 SR(N)=X*SALF+R*CALF                                               SFOR1700
      CALL FREFSH(NS,SX,SR)                                             SFOR1710
C                                                                       SFOR1720
C     CONTROL POINT LOOP                                                SFOR1730
C                                                                       SFOR1740
      DO 21 N=1,NCONPT                                                  SFOR1750
C                                                                       SFOR1760
C     LOCATE POINT IN FUSELAGE SYSTEM                                   SFOR1770
C                                                                       SFOR1780
      XXX=XMOM-XBC(N)                                                   SFOR1790
      CALL STTOIN(XXX,0.0,0.0,XI,ETA,ZETA,DC)                           SFOR1800
      XB=VAR(7)+XI                                                      SFOR1810
      YB=VAR(8)+ETA                                                     SFOR1820
      ZB=VAR(9)+ZETA                                                    SFOR1830
C                                                                       SFOR1840
C     CALCULATE THE BETAS TO BE USED IN THE SOURCE-DOUBLET              SFOR1850
C          VELOCITY CALCULATION FOR ALL AXISYMMETRIC BODIES EXCEPT THE  SFOR1860
C          SEPARATED STORE                                              SFOR1870
C                                                                       SFOR1880
      CALL BVARIA(XB,YB,ZB)                                             SFOR1890
C                                                                       SFOR1900
C     CALCULATE LOCAL WING THICKNESS MACH NUMBER AT THIS XB             SFOR1910
C                                                                       SFOR1920
      IF (XB.GT.XT3.OR.XB.LT.XT4) GO TO 26                              SFOR1930
      FNUMCH=FMXT3+(FMXT4-FMXT3)*(XB-XT3)/(XT4-XT3)                     SFOR1940
      BTSQNU=FNUMCH*FNUMCH-1.0                                          SFOR1950
      BETANU=SQRT(BTSQNU)                                               SFOR1960
      GO TO 27                                                          SFOR1970
   26 FNUMCH=FMACH                                                      SFOR1980
      BETANU=BETA                                                       SFOR1990
      BTSQNU=BETASQ                                                     SFOR2000
   27 CONTINUE                                                          SFOR2010
C                                                                       SFOR2020
C     CALCULATE PERTURBATION VELOCITY FIELD AND RESOLVE INTO STORE      SFOR2030
C          COORDINATE SYSTEM                                            SFOR2040
C                                                                       SFOR2050
      CALL RESVEL(XB,YB,ZB,URES,VRES,WRES)                              SFOR2060
      CALL INTOST(URES,VRES,WRES,UP(N),VP(N),WP(N),DC)                  SFOR2070
   21 CONTINUE                                                          SFOR2080
C                                                                       SFOR2090
C     ADD FREE-STREAM COMPONENTS TO PERTURBATION VELOCITIES             SFOR2100
C     MAKE DIMENSIONLESS BY VSTORE                                      SFOR2110
C     RESULTING VELOCITIES ARE POSITIVE IN -XB, +VB, -ZB DIRECTIONS     SFOR2120
C                                                                       SFOR2130
      DO 22 N=1,NCONPT                                                  SFOR2140
```

```
      UT(N)=VRATS*UP(N)+RVX                                             SFOR2150
      VT(N)=VRATS*VP(N)+RVY                                             SFOR2160
   22 WT(N)=VRATS*WP(N)+RVZ                                             SFOR2170
C                                                                       SFOR2180
C     ADD IN DAMPING TERMS                                              SFOR2190
C                                                                       SFOR2200
      IF (NDAMP.EQ.0) GO TO 24                                          SFOR2210
      DO 23 N=1,NCONPT                                                  SFOR2220
      DUM=(XBC(N)-XMOM)/VSTORE                                          SFOR2230
      VT(N)=VT(N)+VAR(6)*DUM                                            SFOR2240
   23 WT(N)=WT(N)+VAR(5)*DUM                                            SFOR2250
   24 CONTINUE                                                          SFOR2260
C                                                                       SFOR2270
C     CALCULATE SOURCE AND DOUBLET STRENGTHS                            SFOR2280
C                                                                       SFOR2290
      VRATSA=VRATA/VRATS                                                SFOR2300
      BSS=BETAS*BETAS                                                   SFOR2310
      CALL SDSTRN(SXSHLD(NEJSTR))                                       SFOR2320
C                                                                       SFOR2330
C     CALCULATE NORMAL AND SIDE FORCE DISTRIBUTIONS ACTING ON BODY      SFOR2340
C                                                                       SFOR2350
      TA=FMACHS*FMACHS                                                  SFOR2360
      CONA=1.0/(0.7*TA)                                                 SFOR2370
      CONB=0.2*TA                                                       SFOR2380
      COND=-2.0*VY/VSTORE                                               SFOR2390
      CONE=2.0*VZ/VSTORE                                                SFOR2400
      CONC=2.0*SQRT(1.0-(COND*COND+CONE*CONE))                          SFOR2410
      VDAMP=0.0                                                         SFOR2420
      WDAMP=0.0                                                         SFOR2430
      BETAN=SXSHK(2,NEJSTR)/SRSHK(2,NEJSTR)                             SFOR2440
      FMACHN=SQRT(BETAN*BETAN+1.0)                                      SFOR2450
      XINFTY=SXSHLD(NEJSTR)                                             SFOR2460
C                                                                       SFOR2470
C     LOOP OVER X STATIONS                                              SFOR2480
C                                                                       SFOR2490
      DO 50 N=1,NSEG                                                    SFOR2500
      CPX=XCP(N)                                                        SFOR2510
      CPR=RCP(N)                                                        SFOR2520
      RUSREF=CPR/SREF                                                   SFOR2530
      XX=CPX                                                            SFOR2540
      IF (XX.LT.XINFTY) GO TO 56                                        SFOR2550
      BTEAA=BETAA                                                       SFOR2560
      BTEAS=BETAS                                                       SFOR2570
      GO TO 57                                                          SFOR2580
   56 FMLA=FMACHN+(FMACHA-FMACHN)*XX/XINFTY                             SFOR2590
      FMLS=FMACHN+(FMACHS-FMACHN)*XX/XINFTY                             SFOR2600
      BTEAA=SQRT(FMLA*FMLA-1.0)                                         SFOR2610
      BTEAS=SQRT(FMLS*FMLS-1.0)                                         SFOR2620
   57 CONTINUE                                                          SFOR2630
C                                                                       SFOR2640
C     CALCULATE SOURCE AXIAL AND RADIAL VELOCITIES                      SFOR2650
C                                                                       SFOR2660
      SAX=0.0                                                           SFOR2670
      SRAD=0.0                                                          SFOR2680
      DO 51 J=1,NCONPT                                                  SFOR2690
      BAR=BTEAA*CPR                                                     SFOR2700
      SS=SRS(J)                                                         SFOR2710
      ARG=(CPX-XS(J))/BAR                                               SFOR2720
      IF (ARG.LE.1.0) GO TO 52                                          SFOR2730
      CALL SDTRMS                                                       SFOR2740
      SAX=SAX-SS*ACOSH                                                  SFOR2750
   51 SRAD=SRAD+SS*BTEAA*ROOT                                           SFOR2760
   52 CONTINUE                                                          SFOR2770
C                                                                       SFOR2780
C     CALCULATE DOUBLET AXIAL, RADIAL, AND TANGENTIAL VELOCITIES        SFOR2790
C          THE QUANTITIES CALCULATED IN THIS LOOP HAVE NOT BEEN         SFOR2800
C          MULTIPLIED BY SINE OR COSINE OF THETA                        SFOR2810
C                                                                       SFOR2820
      DWAX=0.0                                                          SFOR2830
      DWRAD=0.0                                                         SFOR2840
      DWTAN=0.0                                                         SFOR2850
      DVAX=0.0                                                          SFOR2860
      DVRAD=0.0                                                         SFOR2870
      DVTAN=0.0                                                         SFOR2880
```

Figure C-1(vv)

```
      DO 53 J=1,NCONPT
      BSR=BTEAS*CPR
      TA=0.5*BTEAS*BTEAS
      DW=DSW(J)
      DV=DSV(J)
      ARG=(CPX-XD(J))/BSR
      IF (ARG.LE.1.0) GO TO 54
      CALL SOTRMS
      TB=BTEAS*ROOT
      DWAX=DWAX-DW*TB
      TC=ARG*ROOT
      TD=TA*(TC-ACOSH)
      TE=TA*(TC+ACOSH)
      DWRAD=DWRAD-DW*TE
      DWTAN=DWTAN-DW*TD
      DVAX=DVAX+DV*TB
      DVRAD=DVRAD-DV*TE
   53 DVTAN=DVTAN+DV*TD
   54 CONTINUE
C
C     NONDIMENSIONALIZE SOURCE VELOCITIES BY VSTORE
C
      SAX=SAX/VRATSA
      SRAD=SRAD/VRATSA
C
C     CALCULATE DAMPING TERMS FOR THIS X
C
      IF (NDAMP.EQ.0) GO TO 55
      DUM=(CPX-XMOM)/VSTORE
      WDAMP=VAR(5)*DUM
      VDAMP=VAR(6)*DUM
   55 CONTINUE
C
C     LOOP OVER THETA
C
      XL=CPX/ESTLGC
      DO 60 K=1,NTHETA
      ST=STHETA(K)
      CT=CTHETA(K)
      UXB=SAX+CT*DWAX+ST*DVAX
      RADSD=SRAD+CT*DWRAD+ST*DVRAD
      TANSD=ST*DWTAN+CT*DVTAN
      VYB=RADSD*ST+TANSD*CT+VDAMP
      WZB=RADSD*CT-TANSD*ST+WDAMP
C
C     COMPUTE PRESSURE
C
      VSQ=1.0+UXB*(CONC+UXB)-VYB*(COND-VYB)+WZB*(CONE+WZB)
      TA=1.0+CONB*(1.0-VSQ)
      IF (TA.GE.0.0) GO TO 61
      PPINF=0.0
      GO TO 65
   61 PPINF=TA**3.5
   65 CP(K)=CONA*(PPINF-1.0)
   60 CONTINUE
C
C     INTEGRATE TO OBTAIN DCN/DX AND DCY/DX
C
      DO 62 K=1,NTHETA
   62 FUNC(K)=CP(K)*CTHETA(K)
      CALL SIMSON (NTHETA,FUNC,DTHETA,SUM)
      CNX(N)=-ROSREF*SUM
      DO 63 K=1,NTHETA
   63 FUNC(K)=CP(K)*STHETA(K)
      CALL SIMSON (NTHETA,FUNC,DTHETA,SUM)
      CYX(N)=-ROSREF*SUM
   50 CONTINUE
C
C     INTEGRATE AXIALLY TO OBTAIN FORCES AND MOMENTS
C
      CNSB=0.0
      CYSB=0.0
      CLMSB=0.0
      CLNSB=0.0
      CLLSB=0.0
      DO 70 N=1,NSEG
      DX=DELX
      DXARM=DX*(XMOM-XCP(N))/REFL
      CNSB=CNSB+CNX(N)*DX
      CYSB=CYSB+CYX(N)*DX
      CLMSB=CLMSB+CNX(N)*DXARM
      CLNSB=CLNSB+CYX(N)*DXARM
   70 CONTINUE
      RETURN
      END


      SUBROUTINE SFORC2
C
C     SUBROUTINE TO CALCULATE AERODYNAMIC FORCES AND MOMENTS ON AN
C          ELLIPTIC BODY USING THE THREE-DIMENSIONAL METHOD WITH
C          SOURCE PANELS
C
      LOGICAL SUPERS,SUBSON,IMAGE,SIMAGE,LA(1)
      COMMON /BODCOM/ AMACH,TANO,CX,XCOR(4),YCOR(4),ZCOR(4)
     1   ,XI,YI,ZI,XJ,ZJ,BETA0,BETAL,SUBSON,SUPERS,JJ
      COMMON /BOPTNS/ OPTS(30)
     3   ,REFA,REFD,REFLG,REFX,REFZ,CCTEST,ITMAX,BODL,IZ1(12)
      COMMON /BSWINT/ IP,XPTIP,YPTIP,ZPTIP,DELTIP,DRDXS,YBS,ZBS,BTNOSN
      COMMON /BVELFS/ VX,VY,VZ,VRATS,RVX,RVY,RVZ
      COMMON /CFORCE/ CNX(80),CYX(80),XCP(81),DRDXCP(81),RCP(81)
     *   ,UT(81),VT(81),WT(81)
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *   ,CDC,XMOM,ESTRMX,DX,DELX,VSTORE,ESTLGC
      COMMON /DIMENS/ NX,NR,KX,KR,NXNP,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,I..,IDEL,NTAP7
     *   ,IAR,IAN,IUB,IGB(7),IVR,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IAO,IDO,ISKO,IYIM,IZIM,ISVN,ISKP
     *   ,NRING,IROW(50)
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NHSEGO,NPOLL
      COMMON /LETEM/ FMX13,FMXT4
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1   ,IDEJ,IDF,ID2
      COMMON /NEWFOR/ NTHETA,DTHETA,THETAU(37),STHETA(37),CTHETA(37),
     1 REFL,SREF,NCONPT
      COMMON /NUFLOW/FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,YT1,YT4
      COMMON /PARAM / XMACH,ALPHA,BET2,ALPHAC,PHIM,EM
     1   ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      COMMON /STORIF/ IMFSTR,BTNOSF,RSS,YRN,ZRN,SFAC
      COMMON /STORIM/ IMSTOR,BTNOSE,XWN,YWN,ZWN,IMAGE
      COMMON /WGEOM/ XBWOC,ZBWO,CRW,ZOIMED
      COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)
      COMMON           A(12000)
      EQUIVALENCE    (LA(1),A(1))
C
C  RESTORE GEOMETRY FOR SOURCE PANEL ARRAYS IN BLANK COMMON
C
      REWIND 7
      CALL IOREAD(7,A,NTAP7)
      ESTLGC = REFLG
C
C  DEFINE MACH NUMBER CONSTANTS
C
      AMACH = FMACH
      BETA0 = RETA
      SUPERS = .TRUE.
      SUBSON = .FALSE.
C
C  DEFINE LOCATION OF TEMPORARY ARRAYS IN BLANK COMMON
C
      IUT  = NTAP7+1
      IVT  = IUT+NBODY
      IWT  = IVT+NBODY
```

Figure C-1 (w...

```
      IYIM = IWT+NBODY
      IZIM = IYIM+NBODY
      ISVN = IZIM+NBODY
      ISKP = ISVN+NBODY
      IAN  = ISKP+NBODY
C
C  LOOPS TO COMPUTE VELOCITIES DUE TO OTHER COMPONENTS AT CONTROL POINTS
C  ON STORE.
C
      YY=VAR(8)
      ZZ=VAR(9)-ZBWO
      CALL NUMACH(YY,ZZ)
C
C  CALCULATE LOCATION AT WHICH THE ELLIPTIC STORE NOSE SHOCK WAVE
C  REFLECTS OFF THE WING STRIKES THE STORE BODY
C
      SIMAGE = .TRUE.
      IMSTOR = 0
      IMFSTR = 0
      IF (SIMAGE) CALL ELRFLW (ALPHAC,PHIR,VAR(12))
      IF (SIMAGE) CALL ELRFLB (ALPHAC,PHIR,VAR(12))
      IMAGE = IMSTOR.EQ.1 .OR. IMFSTR.EQ.1
      JSKP = ISKP-1
      DO 120 I=1,NBODY
      JSKP = JSKP+1
120   LA(JSKP) = .TRUE.
C
      JXPT = IXPT-1
      JYPT = IYPT-1
      JZPT = IZPT-1
      JU = IUT-1
      JV = IVT-1
      JW = IWT-1
      DO 170 ISTR=1,KFUS
      KRING = KFORX(ISTR)-1
      KROW  = KRADX(ISTR)-1
      DO 160 J=1,KRING
C
C  CALCULATE BETAS FOR RING ASSUMING ALL PANELS ON RING HAVE SAME BETA
C
      DO 150 I=1,KROW
      JXPT = JXPT+1
      JYPT = JYPT+1
      JZPT = JZPT+1
      JU = JU+1
      JV = JV+1
      JW = JW+1
C
C        LOCATE POINT IN FUSELAGE SYSTEM
      XN = XMOM-A(JXPT)
      CALL STTOIN (XN,A(JYPT),-A(JZPT),XI,ETA,ZETA,DC)
      XB = VAR(7)+XI
      YB = VAR(8)+ETA
      ZB = VAR(9)+ZETA
C
C     CALCULATE THE BETAS TO BE USED IN THE SOURCE-DOUBLET
C          VELOCITY CALCULATION FOR ALL AXISYMMETRIC BODIES EXCEPT THE
C          SEPARATED STORE
C
      CALL BVARIA(XB,YB,ZB)
C
C  CALCULATE LOCAL WING THICKNESS MACH NUMBER AT XB.
C
      IF (XB.GT.XT3 .OR. XB.LT.XT4) GO TO 140
      FNUMCH = FMXT3+(FMXT4-FMXT3)*(XB-XT3)/(XT4-XT3)
      BTSQNU = FNUMCH*FNUMCH-1.0
      BETANU = SQRT(BTSQNU)
      GO TO 145
 140  FNUMCH = FMACH
      BETANU = BETA
      BTSQNU = BETASQ
 145  CONTINUE
C
C  CALCULATE  PERTURBATION VELOCITY FIELD AND RESOLVE INTO STORE
C  COORDINATE SYSTEM
C
      CALL RESVEL (XB,YB,ZB,URES,VRES,WRES)
      CALL INTOST (URES,VRES,WRES,A(JU),A(JV),A(JW),DC)
C
C
C  ADD FREE STREAM TO PERTURBATION VELOCITIES AND MAKE DIMENSIONLESS
C  BY VSTORE.  (POSITIVE IN -XB, +YB, -ZB DIRECTIONS)
C
      A(JU) = -VRATS*A(JU)+RVX
      A(JV) =  VRATS*A(JV)-RVY
      A(JW) = -VRATS*A(JW)+RVZ
150   CONTINUE
160   CONTINUE
170   CONTINUE
C
C  ADD DAMPING CONTRIBUTION
C
      IF (NDAMP.NE.0) CALL VELDMP (A(IUT),A(IVT),A(IWT),A(IXPT),A(IYPT)
     *                            ,A(IZPT),NBODY,XMOM,VSTORE,VAR)
C
C
C  CHECK WHETHER IMAGE EXISTS
C
      IF (.NOT.IMAGE) GO TO 177
C              CHECK FOR NO WING IMAGE
      IF (IMSTOR.LE.0) GO TO 175
C              CHECK FOR FUSELAGE IMAGE
      IF (IMFSTR.LE.0) GO TO 174
C              USE CLOSER OF FUSELAGE AND WING
      RWIM = 2.*ZWN
      RFIM = 2.*SQRT((YBN-YPTIP)**2+(ZBN+ZPTIP)**2)
      IF (RFIM.LT.RWIM) GO TO 175
C              WING IMAGE LOCATION
174   YIMO = YWN
      ZIMO = -ZWN
      YPTO = YWN
      ZPTO = ZWN
      GO TO 176
C              FUSELAGE IMAGE LOCATION
175   YIMO = YBN+2.*(YBS-YBN)
      ZIMO = ZBN+2.*(ZBS-ZBN)
      YPTO = YBN
      ZPTO = ZBN
C  DEFINE GEOMETRY OF IMAGE STORE
C
176   CALL IMAGYZ (A(IYPT),A(IZPT),A(IYIM),A(IZIM),NBODY
     *    ,VAR(12),YIMO,ZIMO,YPTO,ZPTO)
C
C  DEFINE AVERAGE CENTERLINE VELOCITIES FOR MULTI-FIN CONFIGURATION
C
177   IF (NEMP.GT.1) CALL VWAVG (A(IVT),A(IWT),A(IXPT),NRING,IROW)
C
C
C  CALCULATE PANEL SOURCE STRENGTHS INCLUDING SHOCK REFLECTIONS
C
      CALL SDSTN2 (A(IUT),A(IVT),A(IWT),A(ISKP),IMAGE)
      RETURN
      END



      SUBROUTINE SHAPE (X,NS,XE,C,R,DRDX)
C
      SUBROUTINE TO CALCULATE LOCAL BODY RADIUS AND SURFACE SLOPE
C
      DIMENSION XE(7),C(7,7)
      DO 1 K=1,NS
      XL=XE(K)
      J=K
      IF (X.LE.XL) GO TO 2
    1 CONTINUE
    2 R=C(J,1)+X*C(J,5)+X*X*C(J,6)
```

Figure C-1(xx)

```
      ARG=X*X*C(J,2)+X*C(J,3)+C(J,4)
      DRDX=C(J,5)+2.0*X*C(J,6)
      IF (ARG.LE.0.0) RETURN
      R=R+SQRT(ARG)*C(J,7)
      DRDX=DRDX+(2.0*X*C(J,2)+C(J,3))/(2.0*SQRT(ARG))*C(J,7)
      RETURN
      END


      SUBROUTINE SHKLOC(X,R,NS,SX,SR)
C
C     THIS SUBROUTINE LOCATES THE SHOCK WAVE FROM AN AXISYMMETRIC BODY
C          AT COORDINATES Y,Z RELATIVE TO THE BODY
C
      DIMENSION SX(1),SR(1)
C
C     CALCULATE X LOCATION OF SHOCK
C
      DO 1 J=1,NS
      JJ=J
      IF (R.LT.SR(J)) GO TO 2
    1 CONTINUE
      X=SX(NS)+(SX(NS)-SX(NS-1))*(R-SR(NS))/(SR(NS)-SR(NS-1))
      RETURN
    2 JM=JJ-1
      X=SX(JM)+(SX(JJ)-SX(JM))*(R-SR(JM))/(SR(JJ)-SR(JM))
      RETURN
      END


      SUBROUTINE SIMSON(N,F,DX,SUM)
C     SIMPSON RULE SUBROUTINE
      DIMENSION F(101)
      SUM=F(1)+F(N)
      DO 1 I=2,N,2
    1 SUM=SUM+4.0*F(I)
      M=N-2
      DO 2 I=3,M,2
    2 SUM=SUM+2.0*F(I)
      SUM=DX*SUM/3.0
      RETURN
      END


      SUBROUTINE SMARCH (GB,VB,IA,IMAGE)
C
C  MARCHING MATRIX SOLUTION FOR SUPERSONIC FLOW.
C  SOLUTION IS SOLVED RING BY RING ALONG THE BODY ASSUMING NO INFLUENCE
C  OF PANELS ON THE BODY ON RINGS UP STREAM.
C  IF IMAGE=.TRUE.. MODIFICATION TO BOUNDARY CONDITION, VB, DUE TO
C  PRESENCE OF IMAGE BODY IS CALCULATED.
C    JG = STARTING LOCATION OF GB
C    IV = STARTING LOCATION OF VB
C
      LOGICAL IMAGE
      COMMON /DIMENS/ IZDIM(79),NRING,IROW(50)
      COMMON              A(12000)
C
      DIMENSION GB(1),VB(1)
      JG = 0
      CALL IOREAD (9,IROW,NRING)
      DO 160 JB=1,NRING
      JR = IROW(JB)
      DO 150 IB=JB,NRING
      IR = IROW(IB)
      NR = IR*JR
      CALL IOREAD (9,A(IA),NR)
C
C  IF DIAGONAL BLOCK, SOLVE SIMULTANEOUS EQUATIONS
C               A * GB  =  VB
      IF (IR.NE.JR) GO TO 120
      IV = JG
      IF (IR.NE.JR) STOP 002
      CALL PAS002(A(IA),VB(JG+1),IR,1,IR,1)
      DO 100 I=1,IR
      J = I+JG
  100 GB(J) = VB(J)
      GO TO 150
C
C  FOR NON DIAGONAL BLOCKS, SOLUTION OF PRECEEDING RINGS IS KNOWN
C  EXPLICITFLY.  CONTRIBUTION TO SOLUTION IS FORMED BY DIRECT
C  SUBSTITUTION.  MODIFY R.H.S BY SUBTRACTING INFLUENCE FROM FORWARD
C  RINGS.     FORM PRODUCT            VB = VB - A*GB
C
  120 CONTINUE
      IA1 = IA-1
      DO 140 IP=1,IR
      I = IV+IP
      IJ = IP+IA1
      SUM = 0.0
      DO 130 JP=1,JR
      J = JG+JP
      SUM = SUM+A(IJ)*GB(J)
  130 IJ = IJ+IR
      VB(I) = VB(I)-SUM
  140 CONTINUE
C
  150 IV = IV+IR
C
C  COMPUTE INFLUENCE OF IMAGE STORE ON REMAINING PANELS
C               THE FOLLOWING ROUTINE IS CALLED ONLY IN PROGRAM II
C               WHERE THE INFLUENCE OF THE REFLECTED SHOCK IS FELT
C
      IF (IMAGE) CALL IMAGEV (GB,VB,IMAGE,JG,JR)
  160 JG = JG+JR
      RETURN
      END


      SUBROUTINE SOLVUV (U,V,W,A,GB,NBODY,SUPERS)
C
C  ROUTINE TO READ PANEL ON PANEL INDUCED VELOCITY COEFFICIENTS
C  AND COMPUTE VELOCITY COMPONENTS INDUCED AT PANEL CONTROL POINTS
C
      LOGICAL SUPERS
      DIMENSION A(3,1),U(1),V(1),W(1),GB(1)
      COMMON /DIMENS/ IZDIM(79),NRING,IROW(50)
C
C  INITIALIZE VELOCITIES
C
      DO 100 I=1,NBODY
      U(I) = 0.
      V(I) = 0.
  100 W(I) = 0.
C
C   READ VELOCITY COEFFICIENTS BLOCK BY BLOCK
C
      JG = 0
      JS = 1
      DO 150 JB=1,NRING
      JR = IROW(JB)
      IF (SUPERS) JS=JB
      DO 140 IB=JS,NRING
      IR = IROW(IB)
      IF (IB.EQ.JB) IV=JG
      NB = 3*IR*JR
      CALL IOREAD (9,A,NB)
C
C  COMPUTE PRODUCT  U = U+A*GB  FOR BLOCK
C
```

Figure C-1(yy)

```
      DO 130 IP=1,IR
      IJ = IP
      SU = 0.
      SV = 0.
      SW = 0.
      DO 120 JP=1,JR
      J = JP+JG
      SU = SU+A(1,IJ)*GR(J)
      SV = SV+A(2,IJ)*GB(J)
      SW = SW+A(3,IJ)*GR(J)
      IJ = IJ+IR
120   CONTINUE
      I = IP+IV
      U(I) = U(I)+SU
      V(I) = V(I)+SV
130   W(I) = W(I)+SW
140   IV = IV+IR
150   JG = JG+JR
      RETURN
      END


      SUBROUTINE SORPAN(UPM,VPM,WPM)
C
C  COMPUTE THE THREE COMPONENTS OF VELOCITY INDUCED AT A SPECIFIED
C  CONTROL POINT BY A CONSTANT SOURCE DISTRIBUTION ON A
C  QUADRILATERAL PANEL HAVING LONGITUDINAL TAPER AND INCLINED AT AN
C  ANGLE DELTA TO THE FREE STREAM DIRECTION
C  ********************* SUPERSONIC VERSION ***************************
C
      COMMON /BODCOM/ AMACH,TAND,CX,XC(4),YC(4),ZC(4)
     1    ,XI,YI,ZI,XJ,ZJ,BETAO,BETAL,SUBSON,SUPERS,J
      DIMENSION SX(4),SM(4),DX(4),DY(4),DZ(4),D(4),E(4),F(4),G(4),
     1 H(4),XPM(4),YMX(4),ZAX(4),AYM(4),RPM2(4)
C
      REAL NUM
      LOGICAL SUBSON,SUPERS
      DATA EPS,PI/1.E-5,3.14195265/
      IF (SUBSON) GO TO 220
C
      EP2=EPS*EPS
      BT2 = BETAL*BETAL
      BTA = BETAL
      TA=1.-BT2*TAND*TAND
      IF (TA.LT.0.) GO TO 200
      SM(1)=(YC(2)-YC(1))/CX
      SM(3)=(YC(4)-YC(3))/CX
      SM(2)=SM(1)
      SM(4)=SM(3)
      DO 190 I=1,4
      ZC(I)=ZJ-TAND*(XJ-XC(I))
      SSM=SIGN(1.,SM(I))
      TAM=TA-BT2*SM(I)*SM(I)
      IF (ABS(TAM).LE.EPS) TAM=0.
      SAM=SQRT(ABS(TAM))
      SAMD=1./SAM
      CPM=CX*SAM
      DX(I)=XI-XC(I)
      DY(I)=YI-YC(I)
      DZ(I)=ZI-ZC(I)
      IF (ABS(DX(I)).LE.EPS) DX(I)=0.
      IF (ABS(DY(I)).LE.EPS) DY(I)=0.
      IF (ABS(DZ(I)).LE.EPS) DZ(I)=0.
      RPM2(I)=0.
      D2=DX(I)*DX(I)-BT2*(DY(I)*DY(I)+DZ(I)*DZ(I))
      D(I)=0.
C           CHECK FOR POINT LAYING WITHIN MACH CONE
      DXZ=DX(I)-BTA*ABS(DZ(I))
      IF (DXZ.LT.0.) GO TO 170
      IF (D2.GT.0.) D(I)=SQRT(D2)
      XPM(I)=DX(I)-BT2*(SM(I)*DY(I)+TAND*DZ(I))
      YMX(I)=DY(I)-SM(I)*DX(I)
      ZAX(I)=DZ(I)-TAND*DX(I)
      AYM(I)=TAND*DY(I)-SM(I)*DZ(I)
      IF (ABS(XPM(I)).LE.EPS) XPM(I)=0.
      IF (ABS(YMX(I)).LE.EPS) YMX(I)=0.
      IF (ABS(ZAX(I)).LE.EPS) ZAX(I)=0.
      IF (ABS(AYM(I)).LE.EPS) AYM(I)=0.
      IF (I.LE.2) RPM2(1)=YMX(1)**2+ZAX(1)**2-BT2*AYM(1)**2
      RPM2(2)=RPM2(1)
      IF (I.GT.2) RPM2(3)=YMX(3)**2+ZAX(3)**2-BT2*AYM(3)**2
      RPM2(4)=RPM2(3)
      IF (ABS(RPM2(I)).LE.EP2) RPM2(I)=0.
      RPM=SQRT(ABS(RPM2(I)))
      IF (RPM.LE.EPS) RPM=0.
      DPM=SAM*D(I)
      F(I)=0.
      DNOM=-DX(I)*YMX(I)+BT2*DZ(I)*AYM(I)
      FNUM=D(I)*ZAX(I)
      IF (FNUM.EQ.0..AND.DNOM.EQ.0.) GO TO 10
      IF (D(I).EQ.0..OR.ZAX(I).EQ.0.) FNUM=0.
      F(I)=ATAN2(FNUM,DNOM)
      IF (D(I).EQ.0.) F(I)=F(I)*SIGN(1.,ZAX(I))
10    IF (TAM) 100,90,20
20    IF (D(I).EQ.0.) GO TO 70
      IF (RPM.LE.EPS) GO TO 40
      NUM=XPM(I)+DPM
      G(I)=ALOG(NUM/(BTA*RPM))*SAMD
      GO TO 160
40    SX(I)=SIGN(1.,XPM(I))
      IF (I.EQ.1.AND.XPM(1).LT.CPM) GO TO 130
      IF (I.EQ.3.AND.XPM(3).LT.CPM) GO TO 140
50    IF (I.EQ.2) SGN12=SX(1)*SX(2)
      IF (I.EQ.4) SGN34=SX(3)*SX(4)
      IF (XPM(I)) 60,70,80
60    IF (I.EQ.2.AND.SGN12.LT.0.) GO TO 130
      IF (I.EQ.4.AND.SGN34.LT.0.) GO TO 140
      G(I)=-ALOG(ABS(XPM(I)))*SAMD
      GO TO 160
70    G(I)=0.
      GO TO 160
80    G(I)=ALOG(XPM(I))*SAMD
      GO TO 160
90    G(I)=0.
      IF (XPM(I).GT.BTA*RPM) G(I)=D(I)/XPM(I)
      GO TO 160
100   G(I)=0.
      ARG=SIGN(1.,XPM(I))
      IF (RPM.NE.0.) ARG=XPM(I)/(BTA*RPM)
      IF (ARG.GE.1.) GO TO 160
      IF (ARG.LE.-1.) GO TO 110
      IF (D(I).GT.0.) G(I)=ACOS(ARG)*SAMD
      GO TO 160
110   TRM1=(SM(I)*DY(I)+TAND*DZ(I)+ABS(AYM(I))*SAM)/(TAND*TAND+SM(I)**2)
      IF (DX(I).GT.TRM1) GO TO 120
      F(I)=0.
      IF (SSM.GT.0.) F(I)=PI*SIGN(1.,ZAX(I))
      GO TO 160
120   IF (SSM*YMX(I).GE.0.) GO TO 160
      G(I)=PI*SAMD
      GO TO 160
130   G(1)=500.*SAMD
      GO TO 160
140   G(3)=500.*SAMD
160   H(I)=0.
      HARG=-BTA*DY(I)
      IF (D(I).EQ.0..AND.HARG.EQ.0.) GO TO 180
      H(I)=BTA*ATAN2(D(I),HARG)
      GO TO 180
C
C  SET CORNER EQUAL TO ZERO. POINT AHEAD OF MACH CONE
C
170   F(I)=0.
      G(I)=0.
      H(I)=0.
      AYM(I)=0.
```

Figure C-1(zz)

```
      YMX(I)=0.
      ZAX(I)=0.
      XPM(I)=0.
      DPM=0.
      RPM=0.
      RPM2(2)=RPM2(1)
      RPM2(4)=RPM2(3)
180   E(I)=H(I)-BT2*SH(I)*G(I)
190   CONTINUE
C
C  END LOOP OVER FOUR CORNERS.  SUM INFLUENCE OF CORNERS TO CALCULATE
C  U,V,W IN PANEL COORDINATE SYSTEM.
C
      TAD=1./TA
      E14=(E(1)-E(2)-E(3)+E(4))*TAD
      F14=(F(1)-F(2)-F(3)+F(4))*TAD
      G14=G(1)-G(2)-G(3)+G(4)
      R4PI=1./(2.*PI)
      UPM=-R4PI*(E14/BT2+TAND*F14)
      VPM=-R4PI*G14
      WPM=R4PI*(F14+TAND*E14)
      RETURN
C             ERROR CONDITION  - PANEL AHEAD OF MACH CONE
200   WRITE (6,210) J
      WRITE(6,888) AMACH,TAND,CX,XC,YC,ZC,XI,YI,ZI,XJ,ZJ,BETAO,BETAL
888   FORMAT(10G12.5)
      STOP 210
210   FORMAT (1H0,47HERROR - BODY PANEL SLOPE EXCEEDS MACH ANGLE, J=,I4,
     *   63X,12H** SORPAN **)
220   WRITE(6,230)
      STOP 230
230   FORMAT (//10(2H *),27H MACH NUMBER .LT.1.  SORPAN)
      END


      SUBROUTINE SOUTPT
C
C     SUBROUTINE TO OUTPUT FORCE AND MOMENT DATA AND TRAJECTORY DATA
C
      COMMON /BFORCE/ CNSB,CYSB,CLMSB,CLNSB,CLLSB
      COMMON /CFORCE/ CNX(80),CYX(80),XCP(81),DRDXCP(81),RCP(81)
     *   ,UT(81),VT(81),WT(81)
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *   ,CDC,XMOM,ESTRMA,DX,DELX,VSTORE,ESTLGC
      COMMON /EFORC1/ CYEM1,CNEM1,CLLEM1,CLMEM1,CLNEM1
      COMMON /EFORC2/ CYEM2,CNEM2,CLLEM2,CLMEM2,CLNEM2
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NMSEGO,NROLL
      COMMON /JECTOR/ NJECTR,FXS,FYS,FZS,RMX,RMY,RMZ
      COMMON /NEWFOR/ NTHETA,DTHETA,THETAD(37),STHETA(37),CTHETA(37),
     1REFL,SREF,NCONPT
      COMMON /OUTINI/ XNOSEI,YNOSEI,ZNOSEI,XCGI,YCGI,ZCGI,XBASEI,YBASEI,
     1ZBASEI
      COMMON /THRUST/ NTPOLY,TEND(5),TC(5,6),FTHRUS,NTHRUS
      COMMON /TOTFOR/ CSIDE,CNORM,CROLL,CPITCH,CYAW
C
  701 FORMAT(1H1,4X,6HTIME =,F8.5,8H SECONDS)
  702 FORMAT(/5X,29HFORCE AND MOMENT COEFFICIENTS/25X,2HCN,8X,2HCY,7X,
     13HCLM,7X,3HCLN,7X,3HCLL/7X,4HBODY,    9X,5F10.5/
     2   5X,11HEMPENNAGE 1,4X,5F10.5/   5X,11HEMPENNAGE 2,4X,5F10.5/7X,
     35HTOTAL,8X,5F10.5)
  703 FORMAT(/5X,18HLOAD DISTRIBUTIONS              /14X,5HX, FT,6X,3HX/L
     1,5X,6HDCN/DX,4X,6HDCY/DX)
  704 FORMAT(10X,9F10.5)
  705 FORMAT(/5X,67HLOCATION OF STORE IN FUSELAGE COORDINATE SYSTEM, DIM
     1ENSIONS OF FEET/17X,25HRELATIVE TO FUSELAGE NOSE,9X,28HRELATIVE TO
     2 INITIAL POSITION/20X,2HXF,8X,2HYF,8X,2HZF,11X,6HDEL XF,4X,6HDEL Y
     3F,4X,6HDEL ZF)
  706 FORMAT(/5X,81HTRANSLATIONAL VELOCITIES AND ACCELERATIONS OF STORE
     1IN FUSELAGE COORDINATE SYSTEM/10X,27HRELATIVE TO FUSELAGE MOTION/
     215X,3HDXF,7X,3HDYF,7X,3HDZF,6X,4HD2XF,6X,4HD2YF,6X,4HD2ZF)
  707 FORMAT(/5X,75HROTATIONAL VELOCITIES AND ACCELERATIONS OF STORE IN
     1STORE COORDINATE SYSTEM/16X,1HP,9X,1HQ,9X,1HR,7X,4HPDOT,6X,4HQDOT,
     26X,4HRDOT)
  708 FORMAT(/5X,91HSTORE ANGULAR ORIENTATION IN FUSELAGE COORDINATE SYS
     1TEM AND RATES OF CHANGE OF THESE ANGLES/10X,55HANGLES IN DEGREES,
     2RATES OF CHANGE IN RADIANS PER SECOND/15X,3HPSI,6X,5HTHETA,6X,3HPH
     3I,6X,4HDPSI,5X,6HDTHETA,5X,4HDPHI)
  709 FORMAT(10X,4HNOSE,1X,3F10.5,5X,3F10.5)
  710 FORMAT(10X,4HXMOM,1X,3F10.5,5X,3F10.5)
  711 FORMAT(10X,4HBASE,1X,3F10.5,5X,3F10.5)
  712 FORMAT(/5X,15HSTORE THRUST IS,F11.3,7H POUNDS)
  713 FORMAT(/5X,26HEJECTOR FORCES AND MOMENTS,/15X,2HFX,
     1    8X,2HFZ,8X,2HFY,8X,2HMY,8X,2HMZ,8X,2HMX,/9X,6F10.3)
C
C     OUTPUT TIME
C
      WRITE (6,701) TIME
C
C     OUTPUT FORCES AND MOMENTS
C
      WRITE (6,702) CNSB,CYSB,CLMSB,CLNSB,CLLSB
     *              ,CNEM1,CYEM1,CLMEM1,CLNEM1,CLLEM1
     *              ,CNEM2,CYEM2,CLMEM2,CLNEM2,CLLEM2
     *              ,CNORM,CSIDE,CPITCH,CYAW,CROLL
C
C     OUTPUT EJECTOR FORCES AND MOMENTS
C
      IF (NJECTR.GE.1 .AND. NJECTR.LE.3) WRITE(6,713) FXS,FZS
     1   ,FYS,RMY,RMZ,RMX
C
C     OUTPUT STORE THRUST
C
      IF (NTHRUS.EQ.0) GO TO 2
      WRITE (6,712) FTHRUS
    2 CONTINUE
C
C     OUTPUT LOAD AND VELOCITY DISTRIBUTIONS
C
      WRITE (6,703)
      DO 1 J=1,NSEG
      XL=XCP(J)/ESTLGC
    1 WRITE (6,704) XCP(J),XL,CNX(J),CYX(J)
C
C     CALCULATE AND OUTPUT STORE LOCATION (NOSE, MOMENT CENTER,
C          AND BASE) IN FUSELAGE COORDINATE SYSTEM
C
      XXX=XMOM
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XNOSE=VAR(7)+XI
      YNOSE=VAR(8)+ETA
      ZNOSE=VAR(9)+ZETA
      DXN=XNOSE-XNOSEI
      DYN=YNOSE-YNOSEI
      DZN=ZNOSE-ZNOSEI
      XXX=-(ESTLGC-XMOM)
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XBASE=VAR(7)+XI
      YBASE=VAR(8)+ETA
      ZBASE=VAR(9)+ZETA
      DXB=XBASE-XBASEI
      DYB=YBASE-YBASEI
      DZB=ZBASE-ZBASEI
      DXCG=VAR(7)-XCGI
      DYCG=VAR(8)-YCGI
      DZCG=VAR(9)-ZCGI
      WRITE (6,705)
      WRITE (6,709) XNOSE,YNOSE,ZNOSE,DXN,DYN,DZN
      WRITE (6,710) VAR(7),VAR(8),VAR(9),DXCG,DYCG,DZCG
      WRITE (6,711) XBASE,YBASE,ZBASE,DXB,DYB,DZB
C
C     OUTPUT STORE VELOCITIES
C
      WRITE (6,706)
      WRITE (6,704) VAR(1),VAR(2),VAR(3),DVAR(1),DVAR(2),DVAR(3)
C
C     OUTPUT STORE ACCELERATIONS
```

Figure C-1(aaa)

```
C
      WRITE (6,707)
      WRITE (6,704) VAR(4),VAR(5),VAR(6),DVAR(4),DVAR(5),DVAR(6)
C
C     OUTPUT STORE ANGULAR ORIENTATION
C
      PSI=57.29578*VAR(10)
      THA=57.29578*VAR(11)
      PHI=57.29578*VAR(12)
      WRITE (6,708)
      WRITE (6,704) PSI,THA,PHI,DVAR(10),DVAR(11),DVAR(12)
      RETURN
      END



      SUBROUTINE SPECPR(NDAMP,XM,VSTORE,VAR,HEAD)
C
C     VERSION% DEMON2
C
C     THIS SUBROUTINE COMPUTES BERNOULLI PRESSURES AT CONTROL POINTS
C     OF THE CONSTANT U-VELOCITY PANELS ON THE WING OR FIN SURFACES
C     A DESCRIPTION OF THE METHOD AND EQUATIONS HERE MAY BE FOUND
C     IN SECTION 3.6 OF  NASA CR-3122.
C     ASSUMPTIONS
C       ANGULAR RATE ASSUMED SMALL.
C       ALL QUANTITIES RESOLVED INTO BODY COORDINATES
C       IN CALCULATION OF VELOCITY ON BODY AND FINS ONLY INFLUENCE
C         OF STORE SINGULARITIES AND STORE MOTION ARE ACCOUNTED FOR.
C       NEGLIGIBLE BOUYANCY.  NO PARENT AIRCRAFT AND IMAGES
C
C
C
      DIMENSION PRESSA(250),PRESSB(250),PRESSR(250),PRESSL(250)
      DIMENSION VAR(12),HEAD(20)
C
      LOGICAL BODY,DELTA
C
      COMMON /BVEL   / BDU(250),BDV(250),BDW(250)
      COMMON/ICVEL/UTCHK,VTCHK,WTCHK,IIT,IFT,MJ
      COMMON /INTROT/ PHIDIM,THETIT,PHIFR,PHIFL,PHIFU,PHIFD
      COMMON /ONE    / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1  SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2  ,XLF(250),XLR(250),XRF(250),XRR(250),YLC(250),YRC(250),ZLF(250)
     3  ,ZPF(250),ZLR(250),ZRR(250),SNT(100),CST(100),SNT2(100),
     4  CST2(250),XFRIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5  CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6  TOTLR,UVW(3),UCHK,VCHK,WCHK,DT(4),I,IF,II,K,MSWR,MSWL,
     7  MSWU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8  RA,RB,ERATIO,BODY,DELTA
      COMMON/SPCPRS/DLTP(250)
      COMMON/SPSANG/SINALC,COSALC,SINPHI,COSPHI
      COMMON /VRTXV / VVRTX(250),WVRTX(250),NVRTPL,NVRTX,VRTMAX
C
C
C
      NOCPT=1
      NHP1=NHP+1
C
C
C     POINTS IMMEDIATELY ABOVE AND BELOW WING SURFACE TREATED
C     SEPARATELY DUE TO DISCONTINUITY IN INFLUENCE FUNCTION,
C     WHEN Z EQUALS ZERO.
C
C
C
C     WING SURFACES AND BODY INTERFERENCE PANELS CONTRIBUTION
C
      FACTR1=1.428571429/(FMACH*FMACH)
      FACTR2=0.2*FMACH*FMACH
      DTOR = 3.1415926/180.
      DO 807 K=1,NHP
      IF (K.LE.NRP) PHIF=PHIFR*DTOR
      IF (K.GT.NRP) PHIF=PHIFL*DTOR
      VADVRT=VVRTX(K)
C
C     DISCONTINOUS CONTRIBUTION
C     ADD IN CONTRIBUTIONS FROM EXTERNAL VORTICES.
C
C
C     A...ABOVE        B...BELOW
C
      UTOTA=BDU(K)
      VTOTA=BDV(K)+VADVRT
      WTOTA=BDW(K)+WVRTX(K)
      UTOTB=BDU(K)
      VTOTB=BDV(K)+VADVRT
      WTOTB=BDW(K)+WVRTX(K)
C
      II=K
      IF=K
      CALL VELNOR(XCPT(K),YCPT(K),ZCPT(K))
      UTOTB=UTOTB-UCHK
      VTOTB=VTOTB-VCHK
      WTOTB=WTOTB-WCHK
C
      CALL ROTWF (VCHK,WCHK,VLOC,WLOC,PHIF)
      CALL ROTFW (-VLOC,WLOC,VB,WB,PHIF)
      UB = -UCHK
      UTOTB=UTOTB+UB
      VTOTB=VTOTB+VB
      WTOTB=WTOTB+WB
C
C     CONTRIBUTION FROM ALL OTHER AND BODY INTERFERENCE PANELS.
C
      II=1
      IF=NWBP
      CALL VELNOR(XCPT(K),YCPT(K),ZCPT(K))
      UTOTA=UTOTA+UCHK
      VTOTA=VTOTA+VCHK
      WTOTA=WTOTA+WCHK
C
      UTOTB=UTOTB+UCHK
      VTOTB=VTOTB+VCHK
      WTOTB=WTOTB+WCHK
  801 CONTINUE
C
C     ADD CONTRIBUTIONS DUE TO PITCHING, YAWING AND ROLLING MOTIONS
C
      IF (NDAMP.EQ.0) GO TO 830
      CALL VELDMP (UTOTA,VTOTA,WTOTA,XCPT(K),YCPT(K),ZCPT(K),1
     *  ,XM,VSTORE,VAR)
      CALL VELDMP (UTOTB,VTOTB,WTOTB,XCPT(K),YCPT(K),ZCPT(K),1
     *  ,XM,VSTORE,VAR)
  830 CONTINUE
C
      BDUSQ=UTOTA*UTOTA
      BDVSQ=VTOTA*VTOTA
      BDWSQ=WTOTA*WTOTA
      UBAR=UTOTA*COSALC-VTOTA*SINBET+WTOTA*SINALF
      ARG=1.0-FACTR2*(2.0*UBAR+BDUSQ+BDVSQ+BDWSQ)
      PRESSA(K)=-FACTR1
      IF(ARG.GE.TOTLR)PRESSA(K)=FACTR1*(ARG**3.5-1.0)
      BDUSQ=UTOTB*UTOTB
      BDVSQ=VTOTB*VTOTB
      BDWSQ=WTOTB*WTOTB
      UBAR=UTOTB*COSALC-VTOTB*SINBET+WTOTB*SINALF
      ARG=1.0-FACTR2*(2.0*UBAR+BDUSQ+BDVSQ+BDWSQ)
      PRESSB(K)=-FACTR1
      IF(ARG.GE.TOTLR)PRESSB(K)=FACTR1*(ARG**3.5-1.0
  807 CONTINUE
C
C     POINTS IMMEDIATELY TO THE RIGHT AND THE LEFT OF VERTICAL
C     WING SURFACE TREATED SEPARATELY DUE TO DISCONTINUITY IN
C     INFLUENCE FUNCTIONS WHEN Y EQUALS ZERO.
C     ADD IN CONTRIBUTIONS FROM EXTERNAL VORTICES.
C
```

Figure C-1(bbb)

```
C
      IF (MSWU.EQ.0 .AND. MSWD.EQ.0) GO TO 841
C
      DO 813 K=NHP1,NPANLS
      IF (K.GT.NHP .AND. K.LE.N3P) PHIF=PHIFU*DTOR
      IF (K.GT.N3P.AND.K.LE.NPANLS) PHIF=PHIFD*DTOR
      WADVRT=WVRTX(K)
C
      UTOTR=BDU(K)
      VTOTR=BDV(K)+VVRTX(K)
      WTOTR=BDW(K)+WADVRT
      UTOTL=BDU(K)
      VTOTL=BDV(K)+VVRTX(K)
      WTOTL=BDW(K)+WADVRT
      II=K
      IF=K
      CALL VELNOR(XCPT(K),YCPT(K),ZCPT(K))
      UTOTR=UTOTR-UCHK
      VTOTR=VTOTR-VCHK
      WTOTR=WTOTR-WCHK
      CALL ROTWF (VCHK,WCHK,VLOC,WLOC,PHIF)
      CALL ROTFW (-VLOC,WLOC,VR,WR,PHIF)
      UR=-UCHK
      UTOTR=UTOTR+UR
      VTOTR=VTOTR+VR
      WTOTR=WTOTR+WR
C
C     CONTRIBUTION FROM ALL OTHER AND BODY INTERFERENCE PANELS.
      II=1
      IF=NWBP
      CALL VELNOR(XCPT(K),YCPT(K),ZCPT(K))
      UTOTR=UTOTR+UCHK
      VTOTR=VTOTR+VCHK
      WTOTR=WTOTR+WCHK
      UTOTL=UTOTL+UCHK
      VTOTL=VTOTL+VCHK
      WTOTL=WTOTL+WCHK
C
C
C     ADD CONTRIBUTIONS DUE TO  PITCHING, YAWING AND ROLLING MOTIONS
C
      IF (NDAMP.EQ.0) GO TO 835
      CALL VELDMP (UTOTR,VTOTR,WTOTR,XCPT(K),YCPT(K),ZCPT(K),1
     *   ,XM,VSTORE,VAR)
      CALL VELDMP (UTOTL,VTOTL,WTOTL,XCPT(K),YCPT(K),ZCPT(K),1
     *   ,XM,VSTORE,VAR)
  835 CONTINUE
C
      BDUSQ=UTOTR*UTOTR
      BDVSQ=VTOTR*VTOTR
      BDWSQ=WTOTR*WTOTR
      UBAR=UTOTR*COSALC-VTOTR*SINBET+WTOTR*SINALF
      ARG=1.0-FACTR2*(2.0*UBAR+BDUSQ+BDVSQ+BDWSQ)
      PRESSR(K)=-FACTR1
      IF(ARG.GE.TOTLR)PRESSR(K)=FACTR1*(ARG**3.5-1.0)
      BDUSQ=UTOTL*UTOTL
      BDVSQ=VTOTL*VTOTL
      BDWSQ=WTOTL*WTOTL
      UBAR=UTOTL*COSALC-VTOTL*SINBET+WTOTL*SINALF
      ARG=1.0-FACTR2*(2.0*UBAP+BDUSQ+BDVSQ+BDWSQ)
      PRESSL(K)=-FACTR1
      IF(ARG.GE.TOTLR)PRESSL(K)=FACTR1*(ARG**3.5-1.0)
  813 CONTINUE
C
C
C     CALCULATE PRESSURE DIFFERENCES AT CONTROL POINTS
C
  841 CONTINUE
      DO 821 K=1,NPANLS
      IF (K.LE.NHP) GO TO 822
      GO TO  823
  822 DLTP(K)=PRESSR(K)-PRESSA(K)
      GO TO 824
  823 DLTP(K)=PRESSR(K)-PRESSL(K)
  824 CONTINUE
  821 CONTINUE
C
C  CALCULATION OF PRESSURES AT THE CONTROL POINTS OF THE BODY
C  INTERFERENCE PANELS
C
      CALL BDYPR (NDAMP,XM,VSTORE,VAR)
C
C     CALCULATE FIN OR WING LOADINGS DUE TO BERNOULLI LOADING PRESSURES
C
      CALL LOADS (HEAD)
C
C
      RETURN
      END


      SUBROUTINE SPNLD
C
C     VERSIONS DEMON2
C
C     THIS SUBROUTINE COMPUTES SPAN LOAD DISTRIBUTIONS FOR MONOPLANE
C     OR CRUCIFORM WING OR FIN CONFIGURATIONS ONLY.
C NOTES INTERDIGITATED FINS PRESENTLY EXCLUDED.
C     FIN TRAILING EDGE VORTEX STRENGTH AND SPANWISE
C     LOCATION ARE DETERMINED FROM THE LOAD DISTRIBUTIONS.
C     A DESCRIPTION OF THE METHODS AND EQUATIONS PROGRAMMED MAY
C     BE FOUND IN APPENDIX B OF NASA CR-3122 LESS LEADING AND SIDE
C     EDGE FORCE COMPONENTS
C
      DIMENSION SLOAD (20),
     1  VALMAX(5),YMAX(5),VALNUM(5),ZMAX(5),SLP(20)
C
C
      COMMON /ONE   / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRR(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLB(250),ZRB(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,
     7   MSWU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
      COMMON/SPCPRS/DLTPG(250)
      COMMON/SWEEPS/VSWLER(20),VSWTER(20),VSWLEL(20),VSWTEL(20),
     1  VSWLEU(20),VSWTEU(20),VSWLED(20),VSWTED(20),LVSWP,LEFT,FAC,NCWB
     2,ARPNL(250),WIDTH(250)
      COMMON /TEVRT / GAMTE(20),YCG(20),ZCG(20)
      COMMON/THREE/ANGLR,ANGLL,ANGLU,ANGLD,DELR,DELL,DELU,DELD,SREF,HEFL
      COMMON /VRTXV / VVRTX(250),WVRTX(250),NVRTIN,NVRTX,VRTMAX
C
C
C
C
C     CALCULATE SPANWISE LOAD DISTRIBUTIONS
C
      MSWRP=MSWR+1
      MSWLP=MSWL+1
      MSWUP=MSWU+1
      MSWDP=MSWD+1
      CFC=1.0/(4.0*B2)
      TWOR=4.0*B2
C
C     LOOP 542 IS FOR RIGHT AND LEFT HORIZONTAL FINS
C
      BTW=1.0/B2
      J=1
      IL=0
      ANGL=ANGLR
      SINANG=SIN(ANGL)
      COSANG=COS(ANGL)
      NSWANP=MSWRP
```

Figure C-1(ccc'

```
      SIGN=-1.0
      KUL=MSWRP
      JSTART=1
      JEND=NRP
      JTIP=NRP-NCW+1
      TIPCHD=XRB(NRP)-XRF(JTIP)
      IVRT=0
      DO 530 I=1,20
      YCG(I) = 0.0
      ZCG(I) = 0.0
      GAMTE(I) = 0.0
530   CONTINUE
  540 CONTINUE
      I=0
C
C
      SUMFX=0.0
      SUMFT2=0.0
C
C
      SLP(1)=0.0
      DLT=0.
      VALINT=0.0
      CSINT=0.0
      CSMOM=0.0
      NUMEXT=0
      [illegible]=0
      [illegible]T=0
C
C        [illegible] IL=0.....RIGHT FIN
C            IL=1.....LEFT FIN
C
      DO 542 K=2,KUL
      I=I+1
      SWPANG=ABS(ATAN(SWPPLE(J)))
      COSSWP=COS(SWPANG)
      PSIPLE=SWPANG*57.2957795
      SUM1=0.0
      YCHK=YCPT(J)
      YLOC=YCHK*BTW
      IP1=I+1
      IF (IL.EQ.0) IC=I
      IF (IL.EQ.1) IC=I+MSWRP
      DO 541 L=1,NCW
      SUM1=SUM1+FN(J)
      WIDI=1./WIDTH(J)
C
C     WIDTH(J) IS PANEL SPANWISE DIMENSION,ALWAYS POSITIVE.
C     IT IS CALCULATED IN SUBROUTINE LAYOUT.
C
  541 J=J+1
      DELY=WIDTH(J-1)
      FACTOR=WIDI*CFC
C
C     SLOAD....CNC/2B
C     VALINT.....ACCUMULATED VALUE OF THE INTEGRAL (CNC/2*B)*DELY OVER
C     THE SPANWISE DIRECTION.
C
      SLOAD(I)=SUM1*WIDI*CFC
      VALINT=VALINT+(SLOAD(I)*DELY)
      IF (K.NE.KUL) GO TO 525
      YLOC=((B2+RB)/B2)*SIGN
      SLDTIP=0.0
  525 CONTINUE
      IF (I.EQ.1) GO TO 645
      IM1=I-1
      DLT=ABS(YCHK-YCHKBF)
      SLP(I) =(SLOAD(I)-SLOAD(IM1))/DLT
      REFSLP=ABS(SLOAD(1))/(10.0*B2))
      IF (ABS(SLP(I)).LE.REFSLP) SLP(I)=0.0
      IF (I.EQ.2) GO TO 646
      IF (K.NE.KUL) GO TO 522
  521 SLP(NSPANP)=-SLOAD(I)/(DLT/2.0)
      IF (ABS(SLP(NSPANP)).LE.REFSLP) GO TO 646
      SLPCHK=SLP(I)
      IF (SLPCHK.EQ.0.0) SLPCHK=SLP(I-1)
      IF (SLPCHK.GT.0.0.AND.SLP(NSPANP).LT.0.0) GO TO 643
      IF (SLPCHK.LT.0.0.AND.SLP(NSPANP).GT.0.0) GO TO 643
  522 CONTINUE
      IF (SLP(I).EQ.0.0) GO TO 646
      IF (SLPBF.EQ.0.0.AND.I.GE.4) SLPBF=SLP(I-2)
      IF (SLPBF.GT.0.0.AND.SLP(I).LT.0.0) GO TO 643
      IF (SLPBF.LT.0.0.AND.SLP(I).GT.0.0) GO TO 643
      GO TO 646
C
C     VALMAX IS THE EXTREME VALUE OF CNC/2*B
C     ITS VALUE IS TAKEN EQUAL TO THE (I-1)TH VALUE OF CNC/2B
C     LIKEWISE YMAX
C     VALMAX(1) IS THE VALUE OF CNC/2B NEAREST THE ROOT
C
  643 NUMEXT=NUMEXT+1
      ISEQ=NUMEXT+1
      VALMAX(1)=SLOAD(1)
      YMAX(1)=RB*SIGN
      VALMAX(ISEQ  )=SLOAD(IM1)
      YMAX(ISEQ  )=YCHKBF
      JMN=J-NCW-1
      IF (IL.EQ.0) YOUTSD=YRC(JMN)
      IF (IL.EQ.1) YOUTSD=YLC(JMN)
      RSINT=SLOAD(IM1)*(YOUTSD-YMAX(ISEQ))*SIGN
      VALINT=RSINT+(SLOAD(I)*DELY)
      VALNUM(ISEQ)=VLINBF-RSINT
  646 SLPBF=SLP(I)
  645 YCHKBF=YCHK
      VLINBF=VALINT
      IF (K.EQ.KUL.AND.NUMEXT.EQ.0) GO TO 526
      GO TO 527
  526 VALMAX(1)=SLOAD(1)
      YMAX(1)=YCHK
      VALNUM(1)=VALINT
      NOEXT=1
  527 CONTINUE
  542 CONTINUE
C
C
C     FIN TRAILING EDGE VORTICES CALCULATED NEXT- SEE APPENDIX B.(3122)
C     GAMMA.....GAMMA/VINF,POSITIVE COUNTERCLOCKWISE.
C     YCG.......YBAR,MEASURED FROM BODY CENTERLINE.
C
      IF (NOEXT.EQ.0) GO TO 528
      IVRT=IVRT+1
      GAMTE(IVRT )=(SLOAD(1)*TWOB/2.0)*SIGN
      YCG(IVRT)=(RB+(VALNUM(1)/SLOAD(1)))*SIGN
      GO TO 531
  528 CONTINUE
      NLST=NUMEXT+1
      DO 650 ISEQ=1,NLST
      IVRT=IVRT+1
      ISEQP1=ISEQ+1
      IF (ISEQ.EQ.NLST) GO TO 534
      DIFMAX=VALMAX(ISEQP1)-VALMAX(ISEQ)
      GAMTE(IVRT)=-(TWOB/2.0)*DIFMAX*SIGN
      YCG(IVRT)= ((YMAX(ISEQP1)*VALMAX(ISEQP1)-YMAX(ISEQ)*VALMAX(ISEQ))
     1 /DIFMAX)-(VALNUM(ISEQP1)/DIFMAX)*SIGN
      GO TO 533
  534 GAMTE(IVRT)=(TWOB/2.0)*VALMAX(ISEQ)*SIGN
      YCG(IVRT)=YMAX(ISEQ)+(VALINT/VALMAX(ISEQ))*SIGN
  533 CONTINUE
  650 CONTINUE
  531 CONTINUE
C
C
      IF (MSWL.EQ.0) GO TO 549
      IF(IL.NE.0) GO TO 543
C
C     RE-ENTER ABOVE LOOP FOR LEFT HORIZONTAL WING
C
      IL=1
```

Figure C-1(ddd)

```
      KUL=MSWLP
      JSTART=NRP+1
      JEND=NMP
      JTIP=NMP-NCW+1
      TIPCHO=XLB(NMP)-XLF(JTIP)
      ANGL=ANGLL
      SINANG=SIN(ANGL)
      COSANG=COS(ANGL)
      NSPANP=NSPANP+MSWLP
      SIGN=-1.0
      J=NRP+1
      GO TO 540
  543 IF(MSWU.EQ.0) GO TO 549
C
C     LOOP 546 IS FOR UPPER AND LOWER FINS
C     HERE    IL=0....UPPER FIN
C             IL=1.....LOWER FIN
C
C
      BTW=1.0/B2V
      TWOB=4.0*B2V
      KU=MSWUP
      IL=0
      ANGL=ANGLU
      SINANG=SIN(ANGL)
      COSANG=COS(ANGL)
      NSPANP=MSWUP+NSPANP
      SIGN=+1.0
      J=NMP+1
      JSTART=NMP+1
      JEND=N3P
      JTIP=N3P-NCW+1
      TIPCHO=XRB(N3P)-XRF(JTIP)
  544 CONTINUE
      I=0
      SUMFX=0.0
      SUMFT2=0.0
      SLP(1)=0.0
      DLT=0.0
      VALINT=0.0
      CSINT=0.0
      CSMOM=0.0
      NUMEXT=0
      ISEQ=0
      NOEXT=0
      DO 546 K=2,KU
      I=I+1
      SUM1=0.0
      SWPANG=ABS(ATAN(SWPPLE(J)))
      COSSWP=COS(SWPANG)
      PSIPLE=SWPANG*57.2957795
      ZCHK=ZCPT(J)
      ZLOC=ZCHK*BTW
      IP1=I+1
      IC=I+MSWRP+MSWLP+MSWUP
C
C  HERE% IL=0....UPPER FIN
C        IL=1....LOWER FIN
C
      DO 545 L=1,NCW
      SUM1=SUM1+FN(J)
      WIDI=1./WIDTH(J)
  545 J=J+1
      DELZ=WIDTH(J-1)
C
      FACTOR=WIDI*CFC
      SLOAD(I)=SUM1*FACTOR
      VALINT=VALINT+(SLOAD(I)*DELZ)
      IF (K.NE.KU) GO TO 546
      ZLOC=((B2V+RA)/B2V)*SIGN
      SLDTIP=0.0
      IF (K.NE.KUL) GO TO 564
      ZLOC=((B2V+RA)/B2)*SIGN
      SLDTIP=0.0
  564 CONTINUE
C
C     THIS SECTION FOR DETERMINATION OF TRAILING EDGE VORTICITY.
C     SEARCH FOR EXTREMA IN SPAN LOAD DISTRIBUTION, SEE APPENDIX B (3122
C
      IF (I.EQ.1) GO TO 845
      IM1=I-1
      DLT=ABS(ZCHK-ZCHKBF)
      SLP(I) =(SLOAD(I)-SLOAD(IM1))/DLT
      REFSLP=ABS(SLOAD(1)/(10.0*B2V))
      IF (ABS(SLP(I)).LE.REFSLP) SLP(I)=0.0
      IF (I.EQ.2) GO TO 846
      IF (K.EQ.KU) GO TO 821
      GO TO 822
  821 SLP(NSPANP)=-SLOAD(I)/(DLT/2.0)
      IF (ABS(SLP(NSPANP)).LE.REFSLP) GO TO 846
      SLPCHK=SLP(I)
      IF (SLPCHK.EQ.0.0) SLPCHK=SLP(I-1)
      IF (SLPCHK.GT.0.0.AND.SLP(NSPANP).LT.0.0) GO TO 843
      IF (SLPCHK.LT.0.0.AND.SLP(NSPANP).GT.0.0) GO TO 843
  822 CONTINUE
      IF (SLP(I).EQ.0.0) GO TO 846
      IF (SLPBF.EQ.0.0.AND.I.GE.4) SLPBF=SLP(I-2)
      IF (SLPBF.GT.0.0.AND.SLP(I).LT.0.0) GO TO 843
      IF (SLPBF.LT.0.0.AND.SLP(I).GT.0.0) GO TO 843
      GO TO 846
  843 NUMEXT=NUMEXT+1
      ISEQ=NUMEXT+1
      VALMAX(1)=SLOAD(1)
      ZMAX(1)=RA*SIGN
      VALMAX(ISEQ)=SLOAD(IM1)
      ZMAX(ISEQ)=ZCHKBF
      JMN=J-NCW-1
      IF (IL.EQ.0) ZOUTSD=ZRB(JMN)
      IF (IL.EQ.1) ZOUTSD=ZLB(JMN)
      RSINT=SLOAD(IM1)*(ZOUTSD-ZMAX(ISEQ))*SIGN
      VALINT=RSINT+(SLOAD(I)*DELZ)
      VALNUM(ISEQ)=VLINBF-RSINT
  846 SLPBF=SLP(I)
  845 ZCHKBF=ZCHK
      VLINBF=VALINT
      IF (K.EQ.KU.AND.NUMEXT.EQ.0) GO TO 826
      GO TO 827
  826 VALMAX(1)=SLOAD(1)
      ZMAX(1)=ZCHK
      VALNUM(1)=VALINT
      NOEXT=1
  827 CONTINUE
  546 CONTINUE
C
C     FIN TRAILING EDGE VORTICES CALCULATED NEXT.
C     GAMMA.....GAMMA/VINF,POSITIVE COUNTERCLOCKWISE.
C     ZCG.......ZBAR,MEASURED FROM ROC  CENTERLINE.
C
      IF (NOEXT.EQ.0) GO TO 628
      IVRT=IVRT+1
      GAMTE(IVRT)= (SLOAD(1)*TWOB/2.0)*SIGN
      ZCG(IVRT)=(RA+(VALNUM(1)/SLOAD(1)))*SIGN
      GO TO 631
  628 CONTINUE
      NLST=NUMEXT+1
      DO 651 ISEQ=1,NLST
      IVRT=IVRT+1
      ISEQP1=ISEQ+1
      IF (ISEQ.EQ.NLST) GO TO 634
      DIFMAX=VALMAX(ISEQP1)-VALMAX(ISEQ)
      GAMTE(IVRT)=-(TWOB/2.0)*DIFMAX*SIGN
      ZCG(IVRT)= ((ZMAX(ISEQP1)*VALMAX(ISEQP1)-ZMAX(ISEQ)*VALMAX(ISEQ))
     1 /DIFMAX)-(VALNUM(ISEQP1)/DIFMAX)*SIGN
      GO TO 633
  634 GAMTE(IVRT)=(TWOB/2.0)*VALMAX(ISEQ)*SIGN
      ZCG(IVRT)=ZMAX(ISEQ)+(ABS(VALINT/VALMAX(ISEQ)))*SIGN
  633 CONTINUE
  651 CONTINUE
```

Figure C-1(eee)

```
  831 CONTINUE
C
C
C     RE-ENTER ABOVE LOOP FOR LOWER WING
C
      IF(IL.NE.0) GO TO 549
      IL=1
      KU=NSWD+1
      JSTART=N3P+1
      JEND=NPANLS
      ANGL=ANGLD
      SINANG=SIN(ANGL)
      NSPANP=NSPANP+NSWDP
      SIGN=-1.0
      J=N3P+1
      JTIP=NPANLS-NCW+1
      TIPCHD=XLB(NPANLS)-XLF(JTIP)
      GO TO 544
  549 CONTINUE
C
C
      NVRTX = IVRT
      RETURN
      END


      SUBROUTINE STRDAT (INFU,REFAEJ,REFDEJ,BOOLEJ,CIRSTR)
C
C  ROUTINE TO READ ELLIPTIC EJECTED STORE (NEJSTR) DATA INTO LABELED
C  COMMON  AND ORGANIZE REMAINING ELLIPTIC STORE AND NON-CIRCULAR
C  FUSELAGE DATA SEQUENTIALLY IN BLANK COMMON
C
      LOGICAL CIRSTR
      COMMON /BOPTNS/
     1   J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV
     2   ,INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XMLE
     3   ,REFA,REFD,REFL,REFX,REFZ,CCTEST,ITMAX,BOOL,IZ1(12)
      COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT
     *   ,NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5)
     *   ,IXZSYM,NADIM,NADIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7
     *   ,IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB
     *   ,NAG,NAP,NAV,NAS,NASHK,NAFLD,IA0,ID0,ISK0,IZZ(4),NWING,IROW(50)
      COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA
     1  ,IDEJ,IDF,ID2
      COMMON            A(12000)
C
C  READ EJECTED ELLIPTIC STORE DATA INTO LABELED COMMON FIRST
C
      IDEJ = 0
      LASTEJ = 0
      IF (NEJSHP.LE.0) GO TO 90
      IF (CIRSTR) GO TO 90
      REWIND 10
      IF (NEJSHP.GE.1) CALL FRSTRT (10,4,LASTEJ)
      IF (NEJSHP.EQ.2) LASTEJ=0
      IF (NEJSHP.EQ.2) CALL FRSTRT (10,4,LASTEJ)
      REFAEJ = REFA
      REFDEJ = REFD
      BOOLE' = REFL
C
C  SET ASIDE ARRAY SPACE IN BLANK COMMON FOR VB,U,V,W,AND CP
C  OF EJECTED STORE.
C
      IVB = LASTEJ+1
      IU = IVB+NBODY
      IV = IU +NBODY
      IW = IV +NBODY
      ICP = IW+NBODY
      LASTEJ = ICP+NBODY-1
C
C  READ FUSELAGE DATA INTO BLANK COMMON AFTER STORE
C
   90 IDF    = LASTEJ
      LASTF = LASTEJ
      IF (INFU.EQ.2) REWIND 11
      IF (INFU.EQ.2) CALL FRSTRT (11,5,LASTF)
C
C  READ SECOND NON-CIRCULAR STORE SHAPE, IF EXISTS.
C
      ID2    = LASTF
      LASTA = LASTF
C             CHECK FOR NO SECOND SHAPE
      IF (NESHPT.LE.1) GO TO 120
      IF (NEJSHP.GT.1) GO TO 100
C             READ SHAPE # 2
      CALL FRSTRT (10,5,LASTA)
      GO TO 120
C             READ SHAPE # 1
  100 REWIND 10
      CALL FRSTRT (10,6,LASTA)
  120 CONTINUE
C
C  SAVE BLANK COMMON FOR STORE /FUSELAGE DATA ON TAPE7
      REWIND 7
      CALL IOWRIT (7,A,LASTA)
      NTAP7 = LASTA
      RETURN
      END


      SUBROUTINE STTOIN (X,Y,Z,XI,ETA,ZETA,DC)
C     SUBROUTINE TO TRANSFORM FROM STORE TO INERTIAL SYSTEM
C
      DIMENSION DC(3,3)
      XI=X*DC(1,1)+Y*DC(1,2)+Z*DC(1,3)
      ETA=X*DC(2,1)+Y*DC(2,2)+Z*DC(2,3)
      ZETA=X*DC(3,1)+Y*DC(3,2)+Z*DC(3,3)
      RETURN
      END


      SUBROUTINE SWINT(XZ,RZ,KX,XS,RS,XE,C,NS,BLN,RMAX,DRDXS,DRDXB,XB,
     1 RB,INT)
C
C     SUBROUTINE TO  FIND THE INTERSECTION OF A SHOCK WAVE WITH A
C     AXISYMMETRIC BODY
C
C     ARGUMENT LIST IS AS FOLLOWS --
C
C       XZ,RZ - X,R COORDINATES OF SHOCK ORIGIN RELATIVE TO BODY NOSE
C       KX - NUMBER OF XS-RS TABLE VALUES
C       XS,RS - TABLE OF X,R, VALUES LOCATING SHOCK RELATIVE TO SHOCK
C               ORIGIN
C       XE,C,NS - INFORMATION GIVING POLYNOMIAL DESCRIPTION OF BODY,
C                 R/L = F(X/L)
C       BLN - LENGTH OF BODY
C       RMAX - MAXIMUM RADIUS OF BODY
C       DRDXS - DR/DX OF SHOCK WAVE AT INTERSECTION POINT
C       DRDXB - DR/DX OF BODY AT INTERSECTION POINT
C       XB,RB - X,R OF INTERSECION POINT RELATIVE TO BODY ORIGIN
C     INT -  =1 IF SHOCK INTERSECTS BODY,  =0 IF NOT,  =-1  FOR ERR.
C
      DIMENSION XS(1),RS(1),C(7,7),XE(7)
      RNTRP(X,Y,XX,YY,XG)=Y+(XG-X)*(YY-Y)/(XX-X)
C
C     TOL IS TOLERANCE FACTOR.  ITERATION STOPS WHEN ERROR IS LESS THAN
C                        TOL*RMAX
C
      DATA TOL/.01/
C
C     DETERMINE IF SHOCK INTERSECTS BODY
C
```

Figure C-1(fff)

```
C     FIRST CHECK IF RS VALUES GO HIGH ENOUGH.  IF NOT EXTRAPOLATE THE
C
      AERR=TOL*RMAX
      IF(RS(KX).GE.RZ) GO TO 5
C
C     EXTRAPOLATE TABLE
C
      KP=KX+1
      KM=KX-1
      RS(KP)=RZ
      XS(KP)=XS(KM)+(RZ-RS(KM))*(XS(KX)-XS(KM))/(RS(KX)-RS(KM))
      XAX=XS(KP)+XZ
      JM=KX
      GO TO 6
C
C     LOCATE INTERSECTION OF SHOCK WITH BODY AXIS
C
    5 CONTINUE
      DO 3 I=1,KX
      IF(RS(I).LT.RZ) GO TO 3
      J=I
      GO TO 4
    3 CONTINUE
      INT=-1
      RETURN
    4 JM=J-1
      XAX=RNTRP(RS(JM),XS(JM),RS(J),XS(J),RZ)+XZ
    6 IF(XAX.GT.0..AND.XAX.LT.BLN) GO TO 7
C
C     SHOCK MISSES BODY
C
      INT=0
      RETURN
C
C     SHOCK INTERSECTS BODY.  PIN DOWN LOCATION
C
    7 INT=1
      L=JM
    8 XL=(XS(L)+XZ)/BLN
      CALL SHAPE(XL,NS,XE,C,RL,DRDXB)
      RT=RZ-RL*BLN
      IF(RS(L).LT.RT) GO TO 9
      L=L-1
      GO TO 8
C
C     INTERSECTION IS BETWEEN POINTS L AND L+1
C
    9 XL=XS(L)
      RL=RS(L)
      XR=XS(L+1)
      RR=RS(L+1)
   10 XSC=(XL+XR)/2.0
      RSC=RNTRP(XL,RL,XR,RR,XSC)
      XSCL=(XSC+XZ)/BLN
      CALL SHAPE(XSCL,NS,XE,C,R,DRDXB)
      RB=R*BLN
      ERR=RZ-RB-RSC
      IF(ABS(ERR).GT.AERR) GO TO 11
C
C     DONE
C
      DRDXS=(RSC-RS(L))/(XSC-XS(L))
      XB=XSCL*BLN
      RETURN
C
C     HALVE INTERVAL
C
   11 IF(ERR.LT.0.0) GO TO 12
      XL=XSC
      RL=RSC
      INT=INT+1
      GO TO 10
   12 RR=RSC
      XR=XSC
      INT=INT+1
      GO TO 10
      END

      SUBROUTINE SWINTF (XZ,YBN,ZBN,NS,XS,RS,BLN,XHS,RHS,NFUS,IMFSTR
     *     ,KFORX,KRADX,XPT,YPT,ZPT,THET,DELTA,XC)
C
C  ROUTINE TO LOCATE THE INTERSECTION AND PANEL NUMBER WHERE THE
C  STORE NOSE SHOCK STRIKES THE NON-CIRCULAR FUSELAGE.
C  PROCEDURE USED
C    1 - DEFINE R VS X PLANE FROM FUSELAGE AXIS TO STORE NOSE
C           R - FROM FUSELAGE C'WARD,       X - FROM FUSELAGE NOSE BACK
C    2 - FIND INTERSECTION OF SHOCK XS,RS, WITH FUSELAGE AXIS
C    3 - DEFINE INTERSECTION TO RING OF PANELS AT THET
C    4 - COMPUTE INTERSECTION OF LINE THROUGH PAN CP AND XS,RS
C    5 - CONVERT TO FUSELAGE COORDINATES AND PROJECT ONTO LINE
C          BETWEEN FUSELAGE AXIS AND STORE NOSE.
C
C  ARGUMENT LIST
C
C    XZ,RZ - X,R COORDINATES OF SHOCK ORIGIN RELATIVE TO FUSELAGE
C              NOSE
C    XS,RS,NS - TABLE OF X,R VALUES LOCATING SHOCK RELATIVE TO
C                 SHOCK ORIGIN FOR STORE
C    BLN    - LENGTH OF BODY
C    XHS,RHS - X,R OF INTERSECTION POINT RELATIVE TO FUSELAGE ORIGIN
C    IP     - INDEX OF SOURCE PANEL AT SHOCK INTERSECTION
C    IMFSTR - ERROR CODES =1, INTERSECTION, =0, MISSES FUS, =-1, ERROR
C
      DIMENSION XS(10),RS(10),XC(1)
      DIMENSION KFORX(1),KRADX(1),THET(1),DELTA(1),XPT(1),YPT(1),ZPT(1)
      COMMON /XSWINT/ IP,XPTIP,YPTIP,ZPTIP,DELTIP,DRDXS,YHS,ZHS,RTNDSN
      COMMON /CONSTS/ PI,DTOR
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      TWOPI = 2.*PI
C
C  (1) DEFINE R VS X PLANE
C
      RZ = SQRT(YBN*YBN+ZBN*ZBN)
      SNBS = YBN/RZ
      CSBS = ZBN/RZ
C
C  (2) LOCATE INTERSECTION OF SHOCK WITH FUSELAGE AXIS
C
      DO 100 I=2,NS
      IF (RS(I).GE.RZ) GO TO 110
  100 CONTINUE
      I = NS
      WRITE (6,700) RZ
  700 FORMAT (///10X,44HWARNING - XS,RS EXTRAPOLATION - SWINTF * * *
     *  ,5X,4HRZ =,F8.3)
  110 IM = I-1
      XAX = XZ+XS(IM)+(XS(I)-XS(IM))*(RZ-RS(IM))/(RS(I)-RS(IM))
C               DOES SHOCK HITS FUSELAGE
      IF (XAX.GT.0. .AND. XAX.LT.BLN) GO TO 120
C               MISSES BODY - RETURN
      IMFSTR = 0
      RETURN
C
C  IMPROVE ESTIMATE OF RING OF PANELS  OF SHOCK INTERSECTION.
C  USE RADIUS OF FIRST PANEL IN RING AS APPROXIMATION OF RADIUS FOR
C  MACH CONE CORRECTION
C
  120 IX = 0
      IP = 1
C               IX = INDEX OF X-STATION,   IP = INDEX OF PANEL NUMBER
      DO 140 IFU=1,NFUS
      KFUSOR = KFORX(IFU)
      NR     = KRADX(IFU)-1
      IX = IX+1
      DO 130 L=2,KFUSOR
```

Figure C-1(ggg)

```
      IX = IX+1
      XMZ = XAX+BETA*ZPT(IP)
      IF (XMZ.LE.XC(IX)) GO TO 150
      IP = IP+NR
130   CONTINUE
140   CONTINUE
      IP = IP-NR
C
C  LOCATE TWO SHOCK TABLE VALUES WHICH  XMZ LIES BETWEEN
C
150   RMZ = RZ+ZPT(IP)
      DO 160 I=2,NS
      IF (RS(I).GE.RMZ) GO TO 170
160   CONTINUE
      I = NS
170   IM = I-1
C
C  (3) LOCATE PANEL AROUND CIRCUMFERENCE WHOSE ANGLE FROM CONTROL POINT
C  TO STORE NOSE IS CLOSEST TO PERPENDICULAR
C    ANG = POLAR ANGLE FROM CENTERLINE BETWEEN STORE NOSE AND
C          FUSELAGE CONTROL POINT
C
      DIFF = PI
      DO 180 J=1,NR
      ANG = ATAN2(-(YHN-YPT(IP)),-(ZBN+ZPT(IP)))
      IF (ANG.LT.0.) ANG=ANG+TWOPI
      THETIP = THET(IP)
      IF (THETIP.LT.0.) THETIP=THETIP+TWOPI
      DA  = ABS(ANG-THETIP)
      IF (DA.GT.DIFF) GO TO 180
C             SAVE CLOSER VALUE
      DIFF = DA
      IPERP = IP
180   IP = IP+1
      IP = IPERP
      XPTIP = XPT(IPERP)
      YPTIP = YPT(IPERP)
      ZPTIP = ZPT(IPERP)
      DELTIP = DELTA(IPERP)
      THETIP = THET(IPERP)
C
C  INTERSECTION OF STORE NOSE SHOCK TABLE WITH LINE THROUGH PANEL
C  CONTROL POINT IN PLANE OF PANEL IN BODY COORDINATES
   EQUATION OF PANEL$   R = RPT+DRDXB((X-XPT)
      RPT = SQRT(YPTIP**2+ZPTIP**2)
      DRDXB = TAN(DELTIP)
      R1 = RPT-DRDXB*XPTIP
C
C  EQUATION OF SHOCK SEGMENTS   R = RZ-RS(I-1)+DRDXS(X-XS(I-1)-XZ)
      DRDXS = -(RS(I)-RS(IM))/(XS(I)-XS(IM))
      R2 = RZ-RS(IM)-DRDXS*(XS(IM)+XZ)
C             INTERSECTION$   R1+X*DRDXB=R2+X*DRDXS
      XBS = (R2-R1)/(DRDXB-DRDXS)
      RBS = R1+XBS*DRDXB
      NFSTR = 1
C
C  CONVERT RBS TO YBS AND ZBS IN FUSELAGE COORDINATES
C
      DR = DRDXB*(XBS-XPTIP)
      YBS = DR*SIN(THETIP)+YPTIP
      ZBS = -(DR*COS(THETIP)+ZPTIP)
C
C  PROJECT RBS ONTO LINE BETWEEN FUSELAGE CENTER AND STORE CENTER
C
      RBSP = (YBS-YBN)*SNBS+(ZBS-ZBN)*CSBS
      YBS = RBSP*SNBS+YBN
      ZBS = RBSP*CSBS-ZBN
      RBS = SQRT(YBS*YBS+ZBS*ZBS)
      RETURN
      END
```

```
      SUBROUTINE THRCAL (T)
C
C     SUBROUTINE TO CALCULATE THRUST AT TIME=T
C
      COMMON /THRUST/ NTPOLY,TEND(5),TC(5,6),FTHRUS,NTHRUS
C
      DO 1 K=1,NTPOLY
      TE=TEND(K)
      J=K
      IF (T.LE.TE) GO TO 2
    1 CONTINUE
  701 FORMAT(//5X,60H***** TIME GREATER THAN END OF SPECIFIED THRUST TIM
     1E HISTORY)
      WRITE (6,701)
      STOP
    2 FTHRUS=TC(J,1)+T*(TC(J,2)+T*(TC(J,3)+T*(TC(J,4)+T*(TC(J,5)+T*TC(J,
     16)))))
      RETURN
      END
```

```
      SUBROUTINE VELBD2(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED
C     BY THE BODY INTERFERENCE PANEL CORNER POINTS.
C     VELOCITIES UP,VP,WP ARE RETURNED IN THE WING REFERENCE FRAME.
C     THERE ARE TWO COORDINATE SYSTEMS IN USE IN THIS ROUTINE --
C
C     SYSTEM 1 - WING COORDINATE SYSTEM
C                X,U POSITIVE FORWARD
C                Y,V POSITIVE TO RIGHT
C                Z,W POSITIVE DOWN
C
C     SYSTEM 2 - ORIGIN AT VERTEX OF INFLUENCING SEMI-INFINITE TRIANGLE
C     (REQUERED BY SUBROUTINE VELO2)
C                X,U POSITIVE BACK
C                Y,V POSITIVE TO RIGHT IN PLANE OF INFLUENCING PANEL
C                Z,W POSITIVE OUTWARD NORMAL TO INFLUENCING PANEL
C
C     FIELD POINT COORDINATES XX,YY,ZZ, AND VELOCITIES UP,VP,WP
C     ARE IN SYSTEM 1.
C
      LOGICAL PYPNL,FELTA,FELTAI,FELT,UPPER
      COMMON/GRIDCU/XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
     1      SWP(500),SNBP(40),CSBP(40),THTBP(40),DPSET(500)
      COMMON/NUINDX/NRW,NRP,NRB,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1      NCW1,NCP1,NCWB1,NCWS,NCWS1,NLPS1
      COMMON/VELARG/X,Y,Z,U,VV,WV,EM,TLANC,PYPNL,UP,VP,WP
C
C     STATEMENT FUNCTIONS USED FOR COORDINATE AND VELOCITY ROTATIONS.
C
      ROTA(A,B)=A*CS+B*SN
      ROTB(A,B)=A*SN-B*CS
C
      PYPNL=.FALSE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      YI=YY
      ZI=ZZ
      EM=0.0
C
C     OUTER LOOP VARIABLE J IS INDEX OVER CORNERS IN A RING.
C     INNER LOOP VARIABLE I IS CHORDWISE INDEX.
C
      DO 100 J=1,NRB
      JJ=(J-1)*NCWB1+NPTWP
      JROW=J+NRWP
      CS=CSBP(J)
      SN=SNBP(J)
      YW1=YI-YPT(JROW)
```

Figure C-1(hhh)

```
      YW2=-(YI+YPT(JROW))
      ZW=ZI-ZPT(JROW)
      YDIR=ROTA(YW1,ZW)
      ZDIR=ROTB(YW1,ZW)
      YIMG=ROTA(YW2,ZW)
      ZIMG=ROTB(YW2,ZW)
      UPPER=THTBP(J).GT.90..AND.THTBP(J).LE.180.
C
      DO 50 I=1,NCWB1
      IJ=JJ+I
      X=-XI+XPT(IJ)
      IF(X.LT.0.0) GO TO 99
      IF(ABS(DPNET(IJ)).LT.1.0E-10) GO TO 50
      TU=0.0
      TV=0.0
      TW=0.0
      FELT=.FALSE.
      IF(UPPER) GO TO 20
C
C     EQUATIONS FOR CORNERS IN LOWER LEFT QUADRANT
C
      Y=YDIR
      Z=ZDIR
      CALL VELO2(FELTA)
      IF(.NOT.FELTA) GO TO 5
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
      TU=TU-U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
    5 Y=YIMG
      Z=ZIMG
      CALL VELO2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
      TU=TU-U
      TV=TV-V
      TW=TW+W
      FELT = .TRUE.
      GO TO 45
C
C     EQUATIONS FOR CORNERS IN UPPER LEFT QUADRANT.  NOTE THAT
C     INFLUENCE FUNCTIONS ARE MULTIPLIED BY -1/
C
   20 CONTINUE
      Y=-YDIR
      Z=ZDIR
      CALL VELO2(FELTA)
      IF(.NOT.FELTA) GO TO 25
      V=ROTA(-VV,WV)
      W=ROTB(-VV,WV)
      TU=TU+U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   25 Y=-YIMG
      Z=ZIMG
      CALL VELO2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      V=ROTA(-VV,WV)
      W=ROTB(-VV,WV)
      TU=TU+U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
C
C     IF CORNER AND IMAGE CORNER NOT FELT, GO TO NEXT
C     CHORDWISE ROW.  OTHERWISE CALCULATE VELOCITIES AND SUM.
C
   45 CONTINUE
      IF(.NOT.FELT) GO TO 99
      UP=UP+TU*DPNET(IJ)
      VP=VP+TV*DPNET(IJ)
      WP=WP+TW*DPNET(IJ)
C
   50 CONTINUE
C
   99 CONTINUE
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE VELCAL(T,TC,TX,N,XP,Y,ZP,U1,V1,W1,BETAL,
     1 BODYL,SUMK,SUMKD)
C
C SUBROUTINE TO CALCULATE THE VELOCITIES FOR THE FIELD POINT X,Y,Z DUE
C TO THE BODY SINGULARITIES.
C
C     XP,Y,ZP - FIELD POINT COORDINATES WHERE VELOCITY
C               IS TO BE CALCULATED
C     XP,Y,ZP IN BODY SYSTEM
C           XP - POSITIVE FORWARD
C           Y  - POSITIVE RIGHT
C           ZP - POSITIVE DOWN
C     U1,V1,W1 - VELOCITIES ACCORDING TO THE FOLLOWING SIGN CONVENTION
C            U1 - POSITIVE FORWARD
C            V1 - POSITIVE TO RIGHT
C            W1 - POSITIVE DOWN
C
      DIMENSION T(100),TC(100),TX(101)
C
C     TRANSFORM FIELD POINT COORDINATES TO VELCAL SYSTEM
C
      X=-XP
      Z=-ZP
      XFIELD=X
      RFIELD=SQRT(Y*Y+Z*Z)
      THETA=ATAN2(Y,Z)
C
C CALCULATION OF AXIAL, RADIAL AND TANGENTIAL PERTURBATION VELOCITIES
C
      US=0.
      VS=0.
      UD=0.
      VD=0.
      VTD=0.
      DO 110 J=1,N
      BATA=BETAL
      BATASQ=BATA*BATA
      X1=XFIELD-TX(J)
      BR=BATA*RFIELD
      IF(X1.LE.BR) GO TO 115
      XL=X1/BR
      D23=SQRT(XL*XL-1.)
      ACOSH=ALOG(XL+D23)
      U=-ACOSH
      V=BATA*D23
      US=US+T(J)*U
      VS=VS+T(J)*V
      U=V
      XX=XL*D23
      V=-.5*BATASQ*(ACOSH+XX)
      VT=.5*BATASQ*(ACOSH-XX)
      UD=UD+U*TC(J)
      VD=VD+V*TC(J)
  110 VTD=VTD+VT*TC(J)
C
C     ADD VELOCITIES DUE TO SINGULARITIES STARTING AT BODY BASE
C
      X1=XFIELD-BODYL
      BETAA=BATA
      BR=BETAA*RFIELD
```

Figure C-1(iii)

```
      IF (X1.LE.BR) GO TO 115
      XL=X1/BR
      D23=SQRT(XL*XL-1.0)
      ACOSH=ALOG(XL+D23)
      US=US+SUMK*ACOSH
      VS=VS-SUMK*BETAA*D23
      BETAAS=BETAA*BETAA
      UD=UD-BETAA*D23*SUMKD
      VD=VD+0.5*BETAAS*(XL*D23+ACOSH)*SUMKD
      VTD=VTD+0.5*BETAAS*(XL*D23-ACOSH)*SUMKD
C
C TRANSFORMATION OF VELOCITIES INTO ORIGINAL COORDINATE SYSTEM...I.E....
C U1,V1,W1 RATHER THAN U,VR,VTHETA
C
  115 CONTINUE
      COSTH=COS(THETA)
      U1=-US-UD*COSTH
      VR=VD*COSTH+VS
      SINTH=SIN(THETA)
      VTD=VTD*SINTH
      V1=VR*SINTH+VTD*COSTH
      W1=VTD*SINTH-VR*COSTH
      RETURN
      END


      SUBROUTINE VELDMP (UT,VT,WT,XPT,YPT,ZPT,NCONPT,XMOM,VSTORE,VAR)
C
C  ADD DAMPING DUE TO ANGULAR VELOCITIES OF STORE
C
      DIMENSION UT(1),VT(1),WT(1),XPT(1),YPT(1),ZPT(1),VAR(12)
      DO 120 I=1,NCONPT
      XARM = (XPT(I)-XMOM)/VSTORE
      YARM = YPT(I)/VSTORE
      ZARM = ZPT(I)/VSTORE
      UT(I) = UT(I)-VAR(5)*ZARM-VAR(6)*YARM
      VT(I) = VT(I)+VAR(6)*XARM-VAR(4)*ZARM
120   WT(I) = WT(I)+VAR(5)*XARM-VAR(4)*YARM
      RETURN
      END


      SUBROUTINE VELNOR(XX,YY,ZZ)
C
C     VERSIONB DEMON1
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED BY
C     THE WING AND BODY INTERFERENCE PANELS USING SUPERPOSITION SCHEME
C     SUPERPOSITION OF 4 CORNER SOLUTIONS IS USED
C     VELNOR RETURNS VELOCITIES UP,VP,WP IN WING REFERENCE FRAME...
C     BASIC SUPERPOSITION SCHEME I  GIVEN IN AFFDL TR 74-41, FIGURE 4
C     ADDITIONS FOR ARBITRARY DIHEDRAL ARE GIVEN IN APPENDIX A
C     OF NASA CR-3122
C
      LOGICAL DELTA,BODY,IGTNP,UVEQ0
C
      COMMON /BVELO / UVEQ0
      COMMON /INTRO7/ PHIDIH,THETIT,PHIFR,PHIFL,PHIFJ,PHIFU
      COMMON /ONE    / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1    SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2    ,XLF(250),XLB(250),XRF(250),XRB(250),YLC(250),YRC(250),ZLF(250)
     3    ,ZRF(250),ZLB(250),ZRB(250),SNT(100),CST(100),SNT2(100),
     4    CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5    CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6    TOTLR,U,V,W,UP,VP,WP,WBIP,X,Y,Z,M1,IF,II,MJ,MSWR,MSWL,
     7    MSWU,MSWD,NBIP,NCW,NRP,NPW,NRP,N3P,NOCPT,NOUT,NPANLS,NWAP,
     8    RA,RB,ERATIO,BODY,DELTA
C
      COMMON/WRTR/THTI(100),XWLE
C
      DATA PI/3.141592653590/
C
C
C  NOTES
C    NOCPT=0, CALCULATIONS ARE FOR AERODYNAMIC COEFFICIENTS ONLY
C    NOCPT=1, VELOCITY CALCULATIONS ARE USED IN PRESSURE DETERMINATION
C    NOCPT=2, CALCULATIONS ARE USED FOR FIELD POINT VELOCITY
C              DETERMINATION.
C
      DTOR=PI/180.0
C
      UP=0.
      VP=0.
      WP=0.
      X1=XX
      Y1=YY
      Z1=ZZ
C
C     I IS INDEX OF INFLUENCING PANEL
C
      DO 100 I=II,IF
      UVEQ0=NOCPT.EQ.1
      IGTNP= I.GT.NPANLS
      K=I-NPANLS
      MI=I
      TU=0.
      TV=0.
      TW=0.
      EM1=SWPPLE(I)
      EM2=SWPPTE(I)
C
C********************  CORNER 1  ***************************************
C
C
C
C     IF INFLUENCER IS A BODY INTERFERENCE PANEL,SUPERPOSITION IS
C     PERFORMED IN THE WING SYSTEM . SAME FOR INTERDIGITATED FINS.
C     NOTEB BODY INTERFERENCE PANELS IN 2ND. AND 4TH. QUARTERS ACT LIKE
C     WING PANELS WITH NEGATIVE SWEEP
C
      IF (I.LE.NPANLS) GO TO 60
      IF               ((THTI(K).GT.90.0.AND.THTI(K).LE.180.0).OR.
     1(THTI(K).GT.270.0.AND.THTI(K).LE.360.0)) GO TO 15
   60 CONTINUE
      IF (I.LE.NRP)                 PHIF=PHIFR*DTOR
      IF (I.GT.NRP .AND. I.LE.NWP)  PHIF=PHIFL*DTOR
      IF (I.GT.NWP .AND. I.LE.N3P)  PHIF=PHIFJ*DTOR
      IF (I.GT.N3P .AND. I.LE.NPANLS) PHIF=PHIFU*DTOR
      IF (EM1.LT.0.0) GO TO 15
      SLOPE=EM1
      EM=EM1
      X=X1-XLF(I)
      YT=Y1-YLC(I)
      ZT=Z1-ZLF(I)
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      YW=YT
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
C
C     FOR PANELS WITH POSITIVE SWEEP,SUPERPOSITION IS CORNER 1-2-3+4.
C
C     IF INFLUENCER IS A BODY INFLUENCING PANEL, TRANSFORM WING Y,Z
C     COORDINATES INTO BIP COORDINATE SYSTEM,ONLY NEED TO ROTATE SINCE
C     ORIGIN IS ALREADY AT BIP CORNER
C     THEN TRANSFORM VELOCITIES FROM BIP FRAME BACK TO WING,SAME FOR
C     INTERDIGITATED FINS.
C
      IF (IGTNP) CALL ROTWB(YW,ZW,Y,Z,K)
      IF (UVEQ0) UVEQ0=(ABS(Z).LT.TLRNC).AND.(I.NE.MJ)
      CALL VELO
      VR=V
      WR=W
```

Figure C-1(jjj)

```
      VT=V
      WT=W
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU+U
      TV=TV+V
      TW=TW+W
      ICNR=1
      GO TO 10
C
C
C     FOR PANELS WITH NEGATIVE SWEEP,SUPERPOSITION IS CORNER -1+2+3-4
C
C     CORNER 1 FOR M .LT.0.0
C
   15 CONTINUE
      SLOPE=-EM1
      EM=-EM1
      X=XI-XLF(I)
      YT=YI-YLC(I)
      ZT=ZI-ZLF(I)
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      YW=YT
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
      IF (IGTNP)  CALL ROTWB(YW,ZW,Y,Z,K)
      Y=-Y
      IF (UVEQ0) UVEQ0=(ABS(Z).LT.TLRNC).AND.(I.NE.MJ)
      CALL VELO
      VT=-V
      WT=W
      VB=-V
      WB=W
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU-U
      TV=TV-V
      TW=TW-W
      ICNR=1
      GO TO 17
C
C********************  CORNER 2  ****************************************
C
   10 CONTINUE
      X=XI-XRF(I)
      YT=YI-YRC(I)
      ZT=ZI-ZRF(I)
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      YW=YT
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
      IF (IGTNP) CALL ROTWB(YW,ZW,Y,Z,K)
      CALL VELO
      VB=V
      WB=W
      VT=V
      WT=W
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU-U
      TV=TV-V
      TW=TW-W
      ICNR=2
      GO TO 20
C
C     CORNER 2 FOR M .LT. 0
C
   17 CONTINUE
      X=XI-XRF(I)
      YT=YI-YRC(I)
      ZT=ZI-ZRF(I)
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      YW=YT
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
      IF (IGTNP)  CALL ROTWB(YW,ZW,Y,Z,K)
      Y=-Y
      CALL VELO
      VT=-V
      WT=W
      VB=-V
      WB=W
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU+U
      TV=TV+V
      TW=TW+W
      ICNR=2
C
C********************  CORNER 3  ****************************************
C
   20 CONTINUE
      IF (I.LE.NPANLS) GO TO 61
      IF                 ((THTI(K).GT.90.0.AND.THTI(K).LE.180.0).OR.
     1(THTI(K).GT.270.0.AND.THTI(K).LE.360.0)) GO TO 19
   61 IF (EM2.LT.0.0) GO TO 19
      EM=EM2
      SLOPE=EM2
      X=XI-XLB(I)
      YT=YI-YLC(I)
      ZT=ZI-ZLB(I)
      YW=YT
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
      IF (IGTNP) CALL ROTWB(YW,ZW,Y,Z,K)
      CALL VELO
      VB=V
      WB=W
      VT=V
      WT=W
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU-U
      TV=TV-V
      TW=TW-W
      ICNR=3
      GO TO 30
C
C     CORNER 3 FOR M .LT. 0
C
   19 CONTINUE
      SLOPE=-EM2
      EM=-EM2
      X=XI-XLB(I)
      YT=YI-YLC(I)
      ZT=ZI-ZLB(I)
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      YW=YT
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
      IF (IGTNP)  CALL ROTWB(YW,ZW,Y,Z,K)
      Y=-Y
      CALL VELO
      VT=-V
      WT=W
      VB=-V
      WB=W
```

Figure C-1(kkk)

```
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU+U
      TV=TV+V
      TW=TW+W
      ICNR=3
      GO TO 21
C
C******************** CORNER 4 ****************************************
C
   30 CONTINUE
      X=XI-XRB(I)
      YT=YI-YRC(I)
      ZT=ZI-ZRB(I)
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      YW=YT
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
      IF (IGTNP) CALL ROTWB(YW,ZW,Y,Z,K)
      CALL VELO
      VB=V
      WB=W
      VT=V
      WT=W
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU+U
      TV=TV+V
      TW=TW+W
      ICNR=4
      GO TO 40
C
C     CORNER 4 FOR M.LT.0
C
   21 CONTINUE
      X=XI-XRB(I)
      YT=YI-YRC(I)
      ZT=ZI-ZRB(I)
      IF (ABS(X).LE.TLRNC) X=0.0
      IF (ABS(YT).LE.TLRNC) YT=0.0
      IF (ABS(ZT).LE.TLRNC) ZT=0.0
      YW=YT
      ZW=ZT
      IF (.NOT.IGTNP) CALL ROTWF(YT,ZT,Y,Z,PHIF)
      IF (IGTNP)  CALL ROTWB(YW,ZW,Y,Z,K)
      Y=-Y
      CALL VELO
      VT=-V
      WT=W
      VB=-V
      WB=W
      IF (.NOT.IGTNP) CALL ROTFW(VT,WT,V,W,PHIF)
      IF (IGTNP) CALL ROTBW(VB,WB,V,W,K)
      TU=TU-U
      TV=TV-V
      TW=TW-W
      ICNR=4
   40 CONTINUE
C
      IF(NOCPT.EQ.0) GO TO 101
      UP=UP+TU*DELTP(I)
      VP=VP+TV*DELTP(I)
      WP=WP+TW*DELTP(I)
  100 CONTINUE
      UP=UP/CONST
      VP=VP/CONST
      WP=WP/CONST
      RETURN
C
C     UP,VP,WP BELOW ARE FOR SETTING UP INFLUENCE FUNCTIONS FOR USE IN
C     THE INFLUENCE MATRIX
C
  101 UP=TU
      VP=TV
      WP=TW
      RETURN
      END
```

```
      SUBROUTINE VELO
C
      VERSION$DEMON2.
C
C     THIS SUBROUTINE CALCULATES THE INFLUENCE OF THE BASIC
C     SEMI-INFINITE TRIANGLES WHICH ARE UNDER CONSTANT LOADING.
C     MORE CORRECTLY, THEY ARE UNDER CONSTANT U DIFFERENCE
C     THE COORDINATE SYSTEM USED HERE IS THE COORDINATE SYSTEM ASSO-
C     CIATED WITH THE TRIANGLE UNDER CONSIDERATION.
C     SOLUTION SCHEME GIVEN IN AFFDL TR 76-41, APPENDIX 2.
C
C
      LOGICAL UVEQ0
      COMMON /BVELO / UVEQ0
      COMMON /ONE   / DELTP(250),FN(250),PNLC(250),SWPPLE(250),
     1   SWPPTE(250),XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250)
     2   ,XLF(250),XLR(250),XRF(250),XRB(250),YLC(250),YRC(250),ZLF(250)
     3   ,ZRF(250),ZLR(250),ZRR(250),SNT(100),CST(100),SNT2(100),
     4   CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     5   CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,
     6   TOTLR,U,V,W,UCHK,VCHK,WCHK,WBIF,X,YS,ZS,MI,IF,II,MJ,MSWR,MSWL,
     7   MSLU,MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     8   RA,RB,ERATIO,BODY,DELTA
C
      DATA PI/3.141592653590/
C
C
C     STATEMENT FUNCTIONS.
C     PLANAR FORMULATION
C
      F1T(X,Y,Z) = Z*SQRT(XSQ-BTSQ*(YSQ+ZSQ))
      F1B(X,Y,Z) = Y*(EML*Y-X)+EML*ZSQ
      FF2(X,Y,Z) = EML*ALOG((X*EML-BTSQ*Y
     1   + SQRT((X*EML-BTSQ*Y)*(X*EML-BTSQ*Y)
     2   - BTSQ*((Y*EML-X)*(Y*EML-X)+ EMLSQ*ZSQ-BTSQ*ZSQ)))
     4   /(BETA*SQRT((Y*EML-X)*(Y*EML-X) + EMLSQ*ZSQ-BTSQ*ZSQ)))
     6   / SQRT(ABS(EMLSQ-BTSQ))
      FF4(X,Y,Z) = (Y/(YSQ+ZSQ))*SQRT(XSQ-BTSQ*(YSQ+ZSQ))
      FF5(X,Y,Z) = ALOG((X+SQRT(XSQ-BTSQ*(YSQ+ZSQ)))
     1   /(BETA*SQRT(YSQ+ZSQ)))
      FF7(X,Y,Z) = (Z/(YSQ+ZSQ))*SQRT(XSQ-BTSQ*(YSQ+ZSQ))
      FFSON(X,Y,Z) = (SQRT(XSQ-BTSQ*(YSQ+ZSQ)))/(X-BETA*Y)
      FSS(X,Y,Z) = EML*ATAN2(SQRT((BTSQ-EMLSQ)*(XSQ-BTSQ*(YSQ+ZSQ))),
     1    X*EML-BTSQ*Y)  / SQRT(ABS(EMLSQ-BTSQ))
C
C
      Y=YS
      Z=ZS
      TOP=0.0
      BOT=0.0
      EML=EM
      XSQ=X*X
      YSQ=Y*Y
      ZSQ=Z*Z
      BTSQ=BETA*BETA
      EMLSQ=EML*EML
C
C     CHECK FOR SUBSONIC,SONIC OR SUPERSONIC LEADING EDGE
C
      IF (X.LT.0.0) GO TO 50
      IF (X.EQ.0.0.AND.Y.EQ.0.0.AND.Z.EQ.0.0) GO TO 50
      IF(ABS(EML).LT.(100.0*TLRNC)) GO TO 120
      YEDGE=X/EML
      GO TO 121
  120 YEDGE=10.0E+07
```

Figure C-1(111)

```
      GO TO 20
C
C     CHECK FOR SUBSONIC,SONIC, OR SUPERSONIC LEADING EDGE
C
  121 ARG=BTSQ*(1./EMLSQ)
      IF (ABS(ARG-1.0).LT.TLRNC) GO TO 25
      IF(ARG.LT.1.0) GO TO 10
      RBT=SQRT(YSQ+ZSQ)*BETA
      IF(X.GT.RBT) GO TO 63
      GO TO 21
   50 U=0.
      V=0.
      W=0.
      RETURN
C
C     SUPERSONIC LEADING EDGE CASE --
C
   20 CONTINUE
C
C     CHECK IF POINT (X,Y,Z) LIES INSIDE, ON, OR OUTSIDE MACH CONE
C     FROM ORIGIN
C
      RBT=SQRT(YSQ+ZSQ)*BETA
      IF(X.GT.RBT) GO TO 24
   21 IF(Y.LT.0.0.OR.Y.GE.YEDGE) GO TO 50
C
C     POINT LIES OUTSIDE MACH CONE FROM ORIGIN BUT THE POINT LIES
C     INSIDE MACH CONE FROM LEADING EDGE AT SAME Y AS P(X,Y,Z)
C
      YC=X*EML/BTSQ
      IF (Y.LT.YC) GO TO 50
      XLE=Y*SLOPE
      XTRNSF=X-XLE
      IF (ABS(XTRNSF).LE.(100.0*TLRNC)) XTRNSF=0.0
      ZCONE=XTRNSF/(SQRT(BTSQ-EMLSQ))
      IF (ABS(Z).GT.ZCONE) GO TO 50
      IF(ABS(Z)     .LT.TLRNC.OR.(Z)     .GT.0.) GO TO 70
      GO TO 71
   70 F1=PI
      GO TO 74
   71 F1=-PI
   74 F2=-(EML*PI)/SQRT(ABS(BTSQ-EMLSQ))
      IF (ABS(EML).LT.(100.0*TLRNC)) F2=-BETA*PI
      IF (XTRNSF.EQ.0.0.AND.Z.EQ.0.0) F2=F2/2.0
      F4= 0.0
      F5=0.
      F7=0.0
      GO TO 47
C
C     POINT LIES INSIDE MACH CONE FROM ORIGIN
C
   24 CONTINUE
      F2=-BETA*ATAN2(SQRT(X*X-BTSQ*(YSQ+ZSQ)),-BETA*Y)
      GO TO 22
   63 F2=-FSS(X,Y,Z)
      GO TO 22
C
C     SONIC LEADING EDGE CASE--
C
   25 CONTINUE
C
C     CHECK IF POINT(X,Y,Z) LIES ON OR OUTSIDE MACH CONE FROM ORIGIN
C     IF TRUE SET PERTURBATION VELOCITIES EQUAL TO ZERO
C
      RBT=SQRT(YSQ+ZSQ)*BETA
      IF (X.LE.RBT) GO TO 50
      F2=-FFSON(X,Y,Z)
      XTRNSF=X-XLE
      IF (XTRNSF.EQ.0.0.AND.Z.EQ.0.0) F2=F2/2.0
      GO TO 22
C
C     SUBSONIC LEADING EDGE CASE--
C
   10 CONTINUE
C
C     CHECK IF POINT (X,Y,Z) LIES ON OR OUTSIDE MACH CONE
C     IF TRUE SET PERTURBATION VELOCITIES TO ZERO
C
      RBT=SQRT(YSQ+ZSQ)*BETA
      IF(X.LE.RBT) GO TO 50
      ARG2=BETA*SQRT((Y*EML-X)*(Y*EML-X)+EMLSQ*ZSQ-BTSQ*ZSQ)
      IF (ABS(ARG2).LT.TLRNC) GO TO 26
      F2=FF2(X,Y,Z)
      GO TO 22
   26 F2=0.0
   22 CONTINUE
      IF (Y.EQ.0.0.AND.Z.EQ.0.0) GO TO 50
      ARGY=Y-YEDGE
      IF(ABS(ARGY).LT.(100.0*TLRNC).AND.ABS(Z).LT.TLRNC) GO TO 50
      TOP=F1T(X,Y,Z)
      BOT=F1B(X,Y,Z)
      IF(ABS(TOP).GE.TLRNC) GO TO 43
      IF (Y.GT.0.0.AND.Y.LT.YEDGE.AND.ABS(Z)    .LT.TLRNC) GO TO 48
      GO TO 44
   48 F1=PI
      GO TO 46
   43 F1=ATAN2(TOP,BOT)
      IF(ABS(F1).LT.TLRNC) F1= 0.
      GO TO 46
   44 F1=0.0
   46 CONTINUE
      F4= FF4(X,Y,Z)
      F5=FF5(X,Y,Z)
      F7=FF7(X,Y,Z)
C
C     CALCULATE PERTURBATION VELOCITIES U/V,V/V,W/V
C
   47 U=F1
      IF(ABS(EML).LT.(100.0*TLRNC)) GO TO 101
      V=-EML*F1+F7
      W=       (EML*(ABS (1.0-BTSQ/EMLSQ)*F2-F5)-F4)
      GO TO 102
  101 CONTINUE
      V= F7
      W=-F4+F2
  102 CONTINUE
      IF (UVEQ0) U=0.0
      IF (UVEQ0) V=0.0
C
C
      RETURN
      END



      SUBROUTINE VELO2(FELT)
C
C     THIS SUBROUTINE CALCULATES THE INFLUENCE OF A
C     SEMI-INFINITE TRIANGLE UNDER CONSTANT U DIFFERENCE, U+/VINF
C
      LOGICAL PYPNL,FELT,INSIDE
C
      COMMON /CONSTS/ PI,DTOR
      COMMON /FLOW/ALFACR,GAMF,FMACH,RHO,VINF,BETAOL,BTSQOL
      COMMON/NUFLOW/FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4
      COMMON /VELARG/X,YS,ZS,U,V,W,EML,TLRNC,PYPNL,UP,VP,WP
C
C     FUNCTION STATEMENTS FOR COMPUTING F1,F4,F5,F7
C
      FN1(X)=ATAN2(Z*ROOTA1,EML*YZSQ-X*Y)
      FN47(Y)=(Y/YZSQ)*ROOTA1
      FN5(X)=ALOG((X+ROOTA1)/(BETA*SQRT(YZSQ)))
C
      IF (INUMCH.EQ.1.AND..NOT.PYPNL) GO TO 2
      BETA=BETAOL
      BTSQ=BTSQOL
      GO TO 3
```

Figure C-1(mmm)

```
    2 BETA=BETANU
      BTSQ=BTSQNU
    3 CONTINUE
C
      FELT=.TRUE.
      IF(X.GE.TLRNC) GO TO 5
C
C     SET VELOCITIES EQUAL TO ZERO FOR WHATEVER REASON
C
    1 U=0.
      V=0.
      W=0.
      FELT=.FALSE.
      RETURN
C
C     CHECK IF INFLUENCING PANEL IS ON THE PYLON.
C     IF SO, Y=ZS AND Z=-YS
C
    5 IF(PYPNL) GO TO 10
      Y=YS
      Z=ZS
      GO TO 15
   10 Y=ZS
      Z=-YS
   15 YSQ=Y*Y
      ZSQ=Z*Z
      YZSQ=YSQ+ZSQ
      EMLSQ=EML*EML
      ARG1=X*X-BTSQ*YZSQ
      INSIDE=ARG1.GT.TLRNC
C
C     CHECK FOR UNSWEPT LEADING EDGE
C
      IF(ABS(EML).LT.TLRNC) GO TO 40
      YYEDGE=Y-X/EML
C
C     CHECK WHETHER LEADING EDGE IS SUBSONIC,SONIC, OR SUPERSONIC.
C
      STEST=BTSQ-EMLSQ
      IF(ABS(STEST).LT.TLRNC) GO TO 20
      IF(STEST.GT.0.) GO TO 30
C
C************** SUBSONIC LEADING EDGE ********************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF IT LIES OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C
      IF(.NOT.INSIDE) GO TO 1
C
C     INSIDE CONE FROM ORIGIN.
C
C     TEST FOR POSSIBLE SINGULARITIES IN FUNCTIONS.
C
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      IF(ABS(YYEDGE).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      ROOTA1=SQRT(ARG1)
      T1=X*EML-BTSQ*Y
      T2=BTSQ*((Y*EML-X)**2+ZSQ*(EMLSQ-BTSQ))
      F2=(EML/SQRT(EMLSQ-BTSQ))*ALOG((T1+SQRT(T1*T1-T2))/SQRT(T2))
      F1=FN1(X)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
      GO TO 100
C
C************** SONIC LEADING EDGE ***********************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C     VELOCITIES ARE SINGULAR OR INDETERMINATE AT X-AXIS
C     OR LEADING EDGE.  SET U,V, AND W TO ZERO.
C
   20 IF(.NOT.INSIDE) GO TO 1
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      IF(ABS(YYEDGE).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      ROOTA1=SQRT(ARG1)
      F2=ROOTA1/(X-BETA*Y)
      F1=FN1(X)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
      GO TO 100
C
C************** SUPERSONIC LEADING EDGE ******************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO MAKE.
C
   30 IF(INSIDE) GO TO 35
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN.
C     DETERMINE IF IT IS INSIDE CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C
      IF(Y.LT.0.0.OR.YYEDGE.GE.0.0) GO TO 1
      YC=X*EML/BTSQ
      IF (Y.LT.YC) GO TO 1
      XLE=Y*EML
      XTRNSF=X-XLE
      ZCONE=XTRNSF/SQRT(STEST)
      IF (ABS(Z).GT.ZCONE) GO TO 1
C
C     INSIDE MACH CONE FROM LEADING EDGE
C
      F1=PI
      IF(Z.LT.0.0) F1=-PI
      F2=PI*EML/SQRT(BTSQ-EMLSQ)
      F4=0.
      F5=0.
      F7=0.
      GO TO 100
C
C     INSIDE MACH CONE FROM ORIGIN
C     VELOCITIES ARE SINGULAR OR INDETERMINATE AT X-AXIS
C     OR LEADING EDGE.  SET U,V, AND W TO ZERO.
C
   35 CONTINUE
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      IF(ABS(YYEDGE).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      ROOTA1=SQRT(ARG1)
      T3=SQRT(BTSQ-EMLSQ)
      F2=(EML/T3)*ATAN2(T3*ROOTA1,X*EML-BTSQ*Y)
      F1=FN1(X)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
      GO TO 100
C
C************** UNSWEPT LEADING EDGE *********************************
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN.
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO MAKE
C
   40 IF(INSIDE) GO TO 45
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN.
C     DETERMINE IF IT IS INSIDE CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO.
C
      IF(Y.LT.0.0) GO TO 1
      RSTR=ABS(Z)*BETA
      IF(X.LE.RSTR) GO TO 1
C
C     INSIDE MACH CONE FROM LEADING EDGE
C
      F1=PI
```

Figure C-1(nnn)

```
      IF(Z.LT.0.0) F1=-PI
      F2=-BETA*PI
      F4=0.
      F5=0.
      F7=0.
      GO TO 48
C
C     INSIDE MACH CONE FROM ORIGIN
C
   45 CONTINUE
      ROOTA1=SQRT(ARG1)
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      F1=ATAN2(Z*ROOTA1,-X*Y)
      F2=-BETA*ATAN2(ROOTA1,-BETA*Y)
      F4=FN47(Y)
      F5=FN5(X)
      F7=FN47(Z)
C
C     VELOCITIES FOR UNSWEPT LE CASE
C
   48 U=F1
      V=F7
      W=F2-F4
      GO TO 105
C
C*************** CALCULATE PERTURBATION VELOCITIES U,V,W ****************
C
  100 U=F1
      V=-EML*F1+F7
      W=EML*((1.0-BTSQ/EMLSQ)*F2-F5)-F4
C
C     TRANSFORM V AND W FOR PYLON PANELS
C
  105 IF(.NOT. PYPNL) GO TO 110
      TEMP=V
      V=-W
      W=TEMP
  110 CONTINUE
      RETURN
      END


      SUBROUTINE VELOT2(FELT)
C
C     THIS SUBROUTINE CALCULATES THE INFLUENCE OF A SEMI-INFINITE
C     TRIANGLE ASSOCIATED WITH A THICKNESS SOURCE PANEL
C
      LOGICAL PYPNL,INSIDE,FELT
      COMMON /CONSTS/ PI,DTOR
      COMMON /FLOW/ALFACR,GAMF,FMACH,RHO,VINF,BETAOL,BTSQOL
      COMMON/NUFLOW/FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4
      COMMON/VELARG/X,YTH,ZTH,UTH,VTH,WTH,EML,TLRNC,PYPNL,UP,VP,WP
C
      IF (INUMCH.EQ.1.AND..NOT.PYPNL) GO TO 6
      BETA=BETAOL
      BTSQ=BTSQOL
      GO  TO 7
    6 BETA=BETANU
      BTSQ=BTSQNU
    7 CONTINUE
      FELT=.TRUE.
      IF(X.GE.TLRNC) GO TO 3
C
C     SET VELOCITIES TO ZERO FOR WHATEVER REASON
C
    1 UTH=0.0
      VTH=0.0
      WTH=0.0
      FELT=.FALSE.
      RETURN
C
C     CHECK IF THE INFLUENCING PANEL IS ON THE PYLON
```

```
C     IF SO, Y=ZTH AND Z=-YTH
C
    3 IF(PYPNL) GO TO 4
      Y=YTH
      Z=ZTH
      GO TO 5
    4 Y=ZTH
      Z=-YTH
    5 YSQ=Y*Y
      ZSQ=Z*Z
      EMLSQ=EML*EML
      ARG1=X*X-BTSQ*(YSQ+ZSQ)
      INSIDE=ARG1 .GT. TLRNC
C
C     CHECK FOR SPECIAL CASE OF UNSWEPT LEADING EDGE
C
      IF(ABS(EML).LT.TLRNC) GO TO 70
      YEDGE=X/EML
      ARGY=Y-YEDGE
C
C     CHECK WHETHER LEADING EDGE IS SUBSONIC, SONIC, OR SUPERSONIC
C
      STEST=BTSQ-EMLSQ
      IF(ABS(STEST).LT.TLRNC) GO TO 20
      IF(STEST.GT.0.0) GO TO 30
C
C*********** SUBSONIC LEADING EDGE ***********
C
C     DETERMINE IF POINT LIES INSIDE OR OUTSIDE MACH CONE FROM ORIGIN
C     IF IT LIES OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO
C
      IF(.NOT.INSIDE) GO TO 1
C
C     POINT LIES INSIDE MACH CONE FROM ORIGIN
C
      RAD=SQRT(ARG1)
      T1=X*EML-BTSQ*Y
      T2=BETA*SQRT((Y*EML-X)*(Y*EML-X)+ZSQ*(EMLSQ-BTSQ))
C
C     TEST FOR POSSIBLE SINGULARITY IN F1 OR F2
C
      IF(ABS(ARGY).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 1
      F2=(EML/SQRT(EMLSQ-BTSQ))*ALOG((T1+SQRT(T1*T1-T2*T2))/T2)
      IF(ABS(Y).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 13
      F1=ATAN2(Z*RAD,EML*(YSQ+ZSQ)-Y*X)
      F5=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)) )
      GO TO 100
C
C     CASE FOR Y AND Z BOTH SMALL
C
   13 F1=0.0
      VTH=0.0
      GO TO 101
C
C*********** SONIC LEADING EDGE ***********
   20 IF(.NOT.INSIDE) GO TO 1
      RAD=SQRT(ARG1)
      F2=RAD/(X-BETA*Y)
C
C     F1 AND F5 SAME AS IN SUBSONIC LEADING EDGE
C
      IF (ABS(Y) .LT. TLRNC .AND. ABS(Z) .LT. TLRNC) GO TO 21
      F5=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)))
      IF(ABS(ARGY).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 22
      F1=ATAN2(Z*RAD,EML*(YSQ+ZSQ)-Y*X)
      GO TO 100
C
C     Y AND Z BOTH SMALL
C
   21 F1=0.
      VTH = 0.
      GO TO 101
C
C     Z SMALL AND Y CLOSE TO LEADING EDGE
```

Figure C-1(ooo)

```
C
   22 F1=0.
      GO TO 100
C
C*********** SUPERSONIC LEADING EDGE ***********
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO BE MADE
C
   30 IF(INSIDE) GO TO 31
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN
C     DETERMINE IF IT IS INSIDE MACH CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO
C
      IF(Y .LT. 0.0 .OR. Y .GE. YEDGE) GO TO 1
      YC=X*EML/BTSQ
      IF (Y.LT.YC) GO TO 1
      XLE=Y*EML
      XTRNSF=X-XLE
      ZCONE=XTRNSF/SQRT(STEST)
      IF (ABS(Z).GT.ZCONE) GO TO 1
C
C     POINT IS INSIDE MACH CONE FROM LEADING EDGE
C
      F2= PI*EML/SQRT(BTSQ-EMLSQ)
      F1=PI
      IF(Z.LT.0.0) F1=-PI
      F5=0.0
      GO TO 100
C
C     POINT IS INSIDE MACH CONE FROM ORIGIN
C
   31 T3=SQRT(BTSQ-EMLSQ)
      RAD=SQRT(ARG1)
      F2=(EML/T3)*ATAN2(RAD*T3,X*EML-BTSQ*Y)
C
C     USE F1 AND F5 AS IN SUBSONIC LEADING EDGE CASE
C
      IF (ABS(Y) .LT. TLRNC .AND. ABS(Z) .LT. TLRNC) GO TO 32
      F5=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)))
      IF(ABS(ARGY).LT.TLRNC.AND.ABS(Z).LT.TLRNC) GO TO 33
      F1=ATAN2(Z*RAD,EML*(YSQ+ZSQ)-Y*X)
      GO TO 100
C
C     Y AND Z BOTH SMALL
C
   32 F1=0.
      VTH=0.
      GO TO 101
C
C     Z SMALL AND Y CLOSE TO LEADING EDGE
C
   33 F1=0.
      GO TO 100
C
C*********** SPECIAL CASE FOR UNSWEPT LEADING EDGE ***********
C
C     DETERMINE WHETHER POINT LIES INSIDE MACH CONE FROM ORIGIN
C     IF OUTSIDE, THERE IS ONE MORE CHECK TO BE MADF
C
   70 IF(INSIDE) GO TO 72
C
C     POINT IS OUTSIDE MACH CONE FROM ORIGIN
C     DETERMINE IF IT IS INSIDE MACH CONE FROM LEADING EDGE
C     IF OUTSIDE, SET PERTURBATION VELOCITIES TO ZERO
C
      IF(Y .LT. 0.) GO TO 1
      RSTR=ABS(Z)*BETA
      XTEST=X-RSTR
      IF((RSTR .LT. TLRNC .AND. X.LE.0.).OR.XTEST.LE.TLRNC)GO TO 1
C
C     POINT BETWEEN MACH CONE FROM ORIGIN AND LEADING EDGE
C
      UTH=-PI/BETA
      F1=PI
      IF(Z.LT.0.0) F1=-PI
      VTH=0.0
      WTH=F1
      GO TO 102
C
C     POINT IS INSIDE MACH CONE FROM ORIGIN
C
   72 RAD=SQRT(ARG1)
      UTH=-ATAN2(RAD,-BETA*Y)/BETA
      F2=0.0
      IF(ABS(Z) .LT. TLRNC .AND. ABS(Y) .LT. TLRNC)GO TO 73
      F1=ATAN2(Z*RAD,-Y*X)
      F5=ALOG((X+RAD)/(BETA*SQRT(YSQ+ZSQ)))
      VTH=-F5
      WTH=F1
      GO TO 102
   73 VTH=0
      WTH=0
      GO TO 102
C
C     COMPUTE PERTURBATION VELOCITIES
C
  100 VTH=F2-F5
  101 UTH=-F2/EML
      WTH=F1
  102 IF (.NOT. PYPNL) GO TO 103
      TEMP=VTH
      VTH=-WTH
      WTH=TEMP
  103 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VELPP2(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED
C     BY THE PYLON CONSTANT U-VELOCITY PANEL CORNER POINTS.
C     VELPP2 RETURNS VELOCITIES UP,VP,WP IN WING REFERENCE FRAME.
C
      COMMON/GRIDCU/XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
     1      SWP(500),SNBP(40),CSBP(40),THTBP(40),DPNET(500)
      COMMON/NUINDX/NRW,NRP,NRB,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1      NCW1,NCP1,NCWB1,NCWS,NCWS1,NCPS1
      COMMON/PYGEOM/YPL,CENTER
      COMMON/VELARG/X,Y,Z,U,V,W,EM,TLRNC,PYPNL,UP,VP,WP
C
C
      LOGICAL PYPNL,FELTA,FELTA1,FELT,CENTER
C
      PYPNL=.TRUE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      YDIR=YY-YPL
      YIMG=YY+YPL
      ZI=ZZ
C
C     LOOP VARIABLE J IS SPANWISE INDEX, I IS CHORDWISE INDES.
C
      DO 100 J=1,NRP
      JJ=(J-1)*NCP1+NPTW
      JROW=J+NRW
      ZC=-ZI+ZPT(JROW)
C
      DO 50 I=1,NCP1
      IJ=JJ+I
      X=-XI+XPT(IJ)
      IF(X.LT.0.0) GO TO 99
```

Figure C-1(ppp)

```
      IF(ABS(DPNET(IJ)).LT.1.0E-10) GO TO 50
      TU=0.0
      TV=0.0
      TW=0.0
      FELT=.FALSE.
      EM1=SWP(IJ)
      IF(EM1.LT.0.0) GO TO 20
C
C     EM1 .GE. ZERO --- CORNER LEADING EDGE SWEPT BACK
C
      EM=EM1
      Y=YOIR
      Z=-ZC
      CALL VELO2(FELTA)
      IF(.NOT.FELTA) GO TO 10
      TU=TU-U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   10 IF(CENTER) GO TO 45
      Y=YIMG
      CALL VELO2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      TU=TU-U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
      GO TO 45
C
C     EM1 .LT. ZERO --- CORNER LEADING EDGE SWEPT FORWARD.
C     NOTE THAT INFLUENCE FUNCTIONS ARE MULTIPLIED BY -1 WHEN
C     LEADING EDGE HAS NEGATIVE SWEEP.
C
   20 CONTINUE
      EM=-EM1
      Y=YOIR
      Z=ZC
      CALL VELO2(FELTA)
      IF(.NOT.FELTA) GO TO 30
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
   30 IF(CENTER) GO TO 45
      Y=YIMG
      CALL VELO2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
C
C     IF CORNER AND IMAGE CORNER NOT FELT, GO TO NEXT
C     CHORDWISE ROW, OTHERWISE, CALCULATE VELOCITIES AND SUM.
C
   45 CONTINUE
      IF(.NOT.FELT) GO TO 99
      UP=UP+TU*DPNET(IJ)
      VP=VP+TV*DPNET(IJ)
      WP=WP+TW*DPNET(IJ)
C
   50 CONTINUE
C
   99 CONTINUE
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE VELPT2(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED
C     BY THE PYLON THICKNESS PANEL CORNER POINTS.
C     VELPT2 RETURNS VELOCITIES UP,VP,WP IN WING REFERENCE FRAME.
C
      COMMON/GRIDTH/XPTS(1000),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
     1     SWPS(1000),THTNET(1000),DZDX(400)
      COMMON/NUINDX/NRW,NRP,NRA,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1     NCW1,NCP1,NCWB1,NCWS,NCWS1,NCPS1
      COMMON/PYGEOM/YPL,CENTER
      COMMON/VELARG/X,Y,Z,U,V,W,EM,TLRNC,PYPNL,UP,VP,WP
C
C
      LOGICAL PYPNL,FELTA,FELTAI,FELT,CENTER
C
      PYPNL=.TRUE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      YOIR=YY-YPL
      YIMG=YY+YPL
      ZI=ZZ
C
C     LOOP VARIABLE J IS SPANWISE INDEX, I IS CHORDWISE INDES.
C
      DO 100 J=1,NRPS
      JJ=(J-1)*NCPS1+NPTWS
      JROW=J+NRWS
      ZC=-ZI+ZPTS(JROW)
C
      DO 50 I=1,NCPS1
      IJ=JJ+I
      X=-XI+XPTS(IJ)
      IF(X.LT.0.0) GO TO 99
      IF(ABS(THTNET(IJ)).LT.1.0E-10) GO TO 50
      TU=0.0
      TV=0.0
      TW=0.0
      FELT=.FALSE.
      EM1=SWPS(IJ)
      IF(EM1.LT.0.0) GO TO 20
C
C     EM1 .GE. ZERO --- CORNER LEADING EDGE SWEPT BACK
C
      EM=EM1
      Y=YOIR
      Z=-ZC
      CALL VELOT2(FELTA)
      IF(.NOT.FELTA) GO TO 10
      TU=TU-U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
   10 IF(CENTER) GO TO 45
      Y=YIMG
      CALL VELOT2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      TU=TU-U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
      GO TO 45
C
C     EM1 .LT. ZERO --- CORNER LEADING EDGE SWEPT FORWARD.
C     NOTE THAT INFLUENCE FUNCTIONS ARE MULTIPLIED BY -1 WHEN
C     LEADING EDGE HAS NEGATIVE SWEEP.
C
   20 CONTINUE
      EM=-EM1
      Y=YOIR
      Z=ZC
      CALL VELOT2(FELTA)
      IF(.NOT.FELTA) GO TO 30
      TU=TU+U
```

Figure C-1(qqq)

```
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
   30 IF(CENTER) GO TO 45
      Y=YIMG
      CALL VELOT2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
C
C     IF CORNER AND IMAGE CORNER NOT FELT, GO TO NEXT
C     CHORDWISE ROW. OTHERWISE, CALCULATE VELOCITIES AND SUM.
C
   45 CONTINUE
      IF(.NOT.FELT) GO TO 99
      UP=UP+TU*THTNET(IJ)
      VP=VP+TV*THTNET(IJ)
      WP=WP+TW*THTNET(IJ)
C
   50 CONTINUE
C
   99 CONTINUE
  100 CONTINUE
C
      RETURN
      END


      SUBROUTINE VELWP2(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED
C     BY THE WING CONSTANT U-VELOCITY PANEL CORNER POINTS.
C     VELWP2 RETURNS VELOCITIES UP,VP,WP IN WING REFERENCE FRAME.
C
      COMMON/GRIDCU/XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
     1     SWP(500),SNHP(40),CSHP(40),THTHP(40),DPNET(500)
      COMMON/NUINDX/NRW,NRP,NHR,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1     NCW1,NCP1,NCWR1,NCWS,NCWS1,NCPS1
      COMMON/VELARG/X,YV,ZV,U,VV,WV,EM,TLRNC,PYPNL,UP,VP,WP
      COMMON/WGEOM/XBWOC,ZBWO,CRW,ZDIMED
C
C
      LOGICAL PYPNL,FELTA,FELTAI,FELT,ZDIMED
C
C     STATEMENT FUNCTIONS FOR COORDINATE AND VELOCITY ROTATIONS.
C
      ROTA(A,B)=A*CSPH-B*SNPH
      ROTB(A,B)=A*SNPH+B*CSPH
      ROTAC(A,B)=A*CSPH+B*SNPH
      ROTBC(A,B)=-A*SNPH+B*CSPH
C
      PYPNL=.FALSE.
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      YI=YY
      ZI=ZZ
C
C     LOOP VARIABLE J IS SPANWISE INDEX. I IS CHORDWISE INDEX.
C
      DO 100 J=1,NRW
      JJ=(J-1)*NCW1
      CSPH=CPHI(J)
      SNPH=SPHI(J)
      Y1=YI-YPT(J)
      Y2=YI+YPT(J)
      Z=-ZI+ZPT(J)
      IF(ZDIMED) GO TO 5
      YDIR=ROTA(Y1,Z)
      ZDIR=ROTB(Y1,Z)
      YIMG=ROTAC(Y2,Z)
      ZIMG=ROTBC(Y2,Z)
      GO TO 10
    5 YDIR=Y1
      ZDIR=Z
      YIMG=Y2
      ZIMG=Z
   10 CONTINUE
C
      DO 50 I=1,NCW1
      IJ=JJ+I
      X=-XI+XPT(IJ)
      IF(X.LT.0.0) GO TO 99
      IF(ABS(DPNET(IJ)).LT.1.0E-10) GO TO 50
      TU=0.0
      TV=0.0
      TW=0.0
      FELT=.FALSE.
      EM1=SWP(IJ)
      IF(EM1.LT.0.0) GO TO 20
C
C     EM1 .GE. ZERO --- CORNER LEADING EDGE SWEPT BACK
C
      EM=EM1
      YV=-YDIR
      ZV=ZDIR
      CALL VELO2(FELTA)
      IF(.NOT.FELTA) GO TO 15
      V=VV
      W=WV
      IF(ZDIMED) GO TO 12
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   12 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   15 YV=YIMG
      ZV=ZIMG
      CALL VELO2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      V=VV
      W=WV
      IF(ZDIMED) GO TO 17
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   17 CONTINUE
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
      GO TO 45
C
C     EM1 .LT. ZERO --- CORNER LEADING EDGE SWEPT FORWARD.
C     NOTE THAT INFLUENCE FUNCTIONS ARE MULTIPLIED BY -1 WHEN LEADING
C     EDGE HAS NEGATIVE SWEEP.
C
   20 CONTINUE
      EM=-EM1
      YV=YDIR
      ZV=ZDIR
      CALL VELO2(FELTA)
      IF(.NOT.FELTA) GO TO 25
      V=VV
      W=WV
      IF(ZDIMED) GO TO 22
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   22 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
```

Figure C-1(rrr)

```
      FELT=.TRUE.
   25 YV=-YIMG
      ZV=ZIMG
      CALL VELO2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      V=VV
      W=WV
      IF(ZDIMED) GO TO 27
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   27 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
C
C     IF CORNER AND IMAGE CORNER NOT FELT, GO TO NEXT
C     CHORDWISE ROW. OTHERWISE CALCULATE VELOCITIES AND SUM.
C
   45 CONTINUE
      IF(.NOT. FELT) GO TO 99
      UP=UP+TU*DPNET(IJ)
      VP=VP+TV*DPNET(IJ)
      WP=WP+TW*DPNET(IJ)
C
   50 CONTINUE
C
   99 CONTINUE
  100 CONTINUE
C
      RETURN
      END



      SUBROUTINE VELWT2(XX,YY,ZZ)
C
C     THIS SUBROUTINE CALCULATES PERTURBATION VELOCITIES INDUCED
C     BY THE WING THICKNESS PANEL CORNER POINTS.
C     VELWT2 RETURNS VELOCITIES UP,VP,WP IN WING REFERENCE FRAME.
C
      COMMON/GRIDTH/XPTS(1000),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
     1     SWPS(1000),THTNET(1000),DZDX(400)
      COMMON/NUINOX/NRW,NRP,NRB,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,
     1     NCW1,NCP1,NCWB1,NCWS,NCWS1,NCPS1
      COMMON/VELARG/X,YV,ZV,U,VV,WV,EM,TLRNC,PYPNL,UP,VP,WP
      COMMON/WGEOM/XBWOC,ZBWO,CR--,ZDIMED
C
C
      LOGICAL PYPNL,FELTA,FELTAI,FELT,ZDIMED
C
C     STATEMENT FUNCTIONS FOR COORDINATE AND VELOCITY ROTATIONS.
C
      ROTA(A,B)=A*CSPH-B*SNPH
      ROTB(A,B)=A*SNPH+B*CSPH
      ROTAC(A,B)=A*CSPH+B*SNPH
      ROTBC(A,B)=-A*SNPH+B*CSPH
C
      PYPNL=.FALSE
      UP=0.0
      VP=0.0
      WP=0.0
      XI=XX
      YI=YY
      ZI=ZZ
C
C     LOOP VARIABLE J IS SPANWISE INDEX. I IS CHORDWISE INDEX.
C
      DO 100 J=1,NRWS
      JJ=(J-1)*NCWS1
      CSPH=CPHS(J)
      SNPH=SPHS(J)
      Y1=YI-YPTS(J)
      Y2=YI+YPTS(J)
      Z=-ZI-ZPTS(J)
      IF(ZDIMED) GO TO 5
      YDIR=ROTA(Y1,Z)
      ZDIR=ROTB(Y1,Z)
      YIMG=ROTAC(Y2,Z)
      ZIMG=ROTBC(Y2,Z)
      GO TO 10
    5 YDIR=Y1
      ZDIR=Z
      YIMG=Y2
      ZIMG=Z
   10 CONTINUE
C
      DO 50 I=1,NCWS1
      IJ=JJ+I
      X=-XI+XPTS(IJ)
      IF(X.LT.0.0) GO TO 99
      IF(ABS(THTNET(IJ)).LT.1.0E-10) GO TO 50
      TU=0.0
      TV=0.0
      TW=0.0
      FELT=.FALSE.
      EM1=SWPS(IJ)
      IF(EM1.LT.0.0) GO TO 20
C
C     EM1 .GE. ZERO --- CORNER LEADING EDGE SWEPT BACK
C
      EM=EM1
      YV=-YDIR
      ZV=ZDIR
      CALL VELOT2(FELTA)
      IF(.NOT.FELTA) GO TO 15
      V=VV
      W=WV
      IF(ZDIMED) GO TO 12
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   12 CONTINUE
      TU=TU-U
      TV=TV-V
      TW=TW-W
      FELT=.TRUE.
   15 YV=YIMG
      ZV=ZIMG
      CALL VELOT2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      V=VV
      W=WV
      IF(ZDIMED) GO TO 17
      V=ROTA(VV,WV)
      W=ROTB(VV,WV)
   17 CONTINUE
      TU=TU-U
      TV=TV+V
      TW=TW-W
      FELT=.TRUE.
      GO TO 45
C
C      EM1 .LT. ZERO --- CORNER LEADING EDGE SWEPT FORWARD.
C      NOTE THAT INFLUENCE FUNCTIONS ARE MULTIPLIED BY -1 WHEN LEADING
C      EDGE HAS NEGATIVE SWEEP.
C
   20 CONTINUE
      EM=-EM1
      YV=YDIR
      ZV=ZDIR
      CALL VELOT2(FELTA)
      IF(.NOT.FELTA) GO TO 25
      V=VV
      W=WV
      IF(ZDIMED) GO TO 22
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
```

Figure C-1(sss)

```
   22 CONTINUE
      TU=TU+U
      TV=TV-V
      TW=TW+W
      FELT=.TRUE.
   25 YV=-YIMG
      ZV=ZIMG
      CALL VELOT2(FELTAI)
      IF(.NOT.FELTAI) GO TO 45
      V=VV
      W=WV
      IF(ZDIHED) GO TO 27
      V=ROTAC(VV,WV)
      W=ROTBC(VV,WV)
   27 CONTINUE
      TU=TU+U
      TV=TV+V
      TW=TW+W
      FELT=.TRUE.
C
C     IF CORNER AND IMAGE CORNER NOT FELT, GO TO NEXT
C     CHORDWISE ROW, OTHERWISE CALCULATE VELOCITIES AND SUM.
C
   45 CONTINUE
      IF(.NOT. FELT) GO TO 99
      UP=UP+TU*THTNET(IJ)
      VP=VP+TV*THTNET(IJ)
      WP=WP+TW*THTNET(IJ)
C
   50 CONTINUE
C
   99 CONTINUE
  100 CONTINUE
C
      RETURN
      END


      FUNCTION VNORM (I,U,V,W,THET,DELTA)
C
C FUNCTION ROUTINE TO COMPUTE THE I*TH VELOCITY NORMAL TO A SURFACE
C AT THE ORIENTATIONS THET AND DELTA FROM VELOCITIES (U,V,W)
C
      DIMENSION U(1),V(1),W(1),THET(1),DELTA(1)
      VNORM = (W(I)*COS(THET(I))-V(I)*SIN(THET(I)))*COS(DELTA(I))
     *   -U(I)*SIN(DELTA(I))
      RETURN
      END


      SUBROUTINE VOTEX(NV,XV,YV,GV,VV,WW)
C
C  THIS SUBROUTINE COMPUTES THE PERTURBATION VELOCITY COMPONENTS AT THE
C     VORTEX LOCATIONS ACCOUNTING FOR MUTUAL EFFECTS AND THE PRESENCE
C     OF A WINGED ELLIPTICAL CROSS SECTION
C
      DIMENSION XV(30),YV(30),GV(30),VV(30),WW(30)
      DIMENSION VX(30),VY(30),G(30)
      COMMON/BLK1/BY,AZ,R,S,PI,CI
      COMMON/COM1/AA,BB,RR
      COMMON/COM2/SIG2,H2
      COMPLEX CI,TO,Z,SO,DSDT,DTDS,D2SDT,V1,V2,V3,VEL,TAU,SI,SOB,SIB
     2,DSDZ,DZDS,D2SDZ2
      EXTERNAL Z,DSDZ,DZDS,D2SDZ2
      NN=NV-1
      AA=BY*BY
      BB=AZ*AZ
      APB=BY*AZ
      APB2=APB*APB
      R1=S+SQRT(S*S-AA+BB)
      H=0.25*(R1+APB2/R1)
      H=R+R
      H2=H*H
      SIG=0.5*APB
      SIG2=SIG*SIG
      RR=R*R
      DO 100 I=1,NV
      XO=XV(I)
      YO=YV(I)
      GO=GV(I)
      K1=0
      DO 99 J=1,NV
      IF(J.EQ.I)GO TO 99
      K1=K1+1
      VX(K1)=XV(J)
      VY(K1)=YV(J)
      G(K1)=GV(J)
   99 CONTINUE
      TO=CMPLX(XO,YO)
      SO=Z(TO)
      SOB=CONJG(SO)
      DSDT=DSDZ(SO)
      DTDS=DZDS(SO)
      D2SDT=D2SDZ2(SO)
      V1=-GO*DTDS*D2SDT/2.
      V2=GO*DSDT*SOB/(SOB*SO-RR)
      V3=CMPLX(0.,0.)
      DO 3 K=1,NN
      TAU=CMPLX(VX(K),VY(K))
      SI=Z(TAU)
      SIB=CONJG(SI)
    3 V3=V3-G(K)*(1./(SO-SI)-SIB/(SO*SIB-RR))*DSDT
      VEL=CI*(V1+V2+V3)/(PI+PI)
      VV(I)=REAL(VEL)
      WW(I)=-AIMAG(VEL)
  100 CONTINUE
      RETURN
      END


      SUBROUTINE VPATH (NOUTG,NVRTX,VRTMAX)
C
C  ROUTINE TO ORGANIZE DATA FOR CALL TO VPATHL FOR MULTI-FIN CONFIGS.
C
C  NOTES  ROUTINE IS RESTRICED TO SINGLE SEGMENT ELLIPTIC BODIES
C         BY TABLE XC.
C
      LOGICAL LPRT,LFIN
      DIMENSION NVV(20)
      COMMON  / BFORC / FROD(5,5),XBOD(5),IBOD(5),LFIN(5),NBOD
      COMMON /BVELFS/ VX,VY,VZ,VRATS,RVX,RVY,RVZ
      COMMON /DIMENS/ IDIM(130)
      COMMON /VEL    / VXAV,WXAV,USTOR,NOUT,NV,NS,NF
      COMMON              A(12000)
      EQUIVALENCE (IDIM(23),IXC)
      DATA ES/0.001/
C
C  FIND STARTING AND ENDING PANEL X-STATIONS
C
      IXP = 0
      NIP = 0
      DO 100 I=1,NBOD
      IF (LFIN(I).AND.IXP.EQ.0) GO TO 120
  100 CONTINUE
  120 IXP = IBOD(I)
      IF (LFIN(I+1))  GO TO 999
      IF (LFIN(I+2)) NIP=IBOD(I+1)-IXP+1
      IXP = IXP+IXC-1
      DO 140 I=1,NIP
  140 NVV(I) = NVRTX
      USTOR = RVX
      NOUT = NOUTG
```

Figure C-1(ttt)

```
      LPRT = NOUTG.GT.0
      NCP = 0
      CALL VPATHL (A(IXP),NVV,NIP,ES,VRTMAX,LPRT
     *   ,XCP,YCP,ZCP,VCP,WCP,NCP)
999   RETURN
      END


      SUBROUTINE VPATHL (XIP,NVV,NIP,ES,VRTMAX,LPRT
     *   ,XCP,YCP,ZCP,VCP,WCP,NCP)
C
C        THIS PROGRAM COMPUTES THE PATHS AND VORTEX INDUCED CROSSFLOW
C   VELOCITIES AT SPECIFIED FIELD POINTS FOR A SET OF VORTICES IN THE
C   PRESENCE OF A BODY IN A NON-UNIFORM FLOW FIELD.
C     SLENDER BODY THEORY IS USED IN THE COMPUTATION OF THE CROSSFLOW
C     VELOCITIES.  THE BODY IS ASSUMED TO HAVE ELLIPTICAL CROSS SECTION.
C
C
C  THE COORDINATE SYSTEM USED HERE IS THE BODY COORDINATE SYSTEM
C  WITH THE X-AXIS ALONG THE BODY CENTER-LINE STARTING AT THE NOSE TIP.
C     Y-AXIS TO THE RIGHT WHEN LOOKING FORWARD,Z-AXIS UP.
C
C
C  ********NOTES BODYS ONLY - NO LIFTING SURFACES********************
C
C
      LOGICAL LPRT
      DIMENSION XO(40),XIP(20),WK(120)
      DIMENSION VXP(20,20,2),NVV(20)
      DIMENSION XCP(1),YCP(1),ZCP(1),VCP(1),WCP(1)
C
      COMMON /TEVRT / G(20),VX(20),VY(20)
      COMMON /VEL   / VXAV,WXAV,USTCY,NOUT,NV,NS,NF
C
  701 FORMAT (////5(1H*),61HSUBROUTINE DASCRU CAN NOT OBTAIN VALUE FOR I
     1NTEGRATION STEP H,5(1H*)/////)
C
C
C  .. GEOMETRY ..
C
C
C        NIP= NO. OF X- STATIONS PRINTED IN OUTPUT
C
C
C        ES = PERMISSIBLE RELATIVE ERROR IN THE PATH INTEGRATION SCHEME
C
      NF = 0
      IF (VRTMAX.EQ.0.0) VRTMAX=0.35
C
C  .. X-STATIONS,INTEGRATION INFO. - INITIAL VORTEX POSITIONS AND STRENG
C
C        NVV = TOTAL NUMBER OF VORTICES AT EACH X-STATION
C
C
      NV=NVV(1)
C
C     HERE....VX(I)..Y-COORDINATE
C             VY(I)..Z-COORDINATE
C             G(I)...GAMMA/VINF,COUNTERCLOCKWISE POSITIVE.
C                    WHEN VIEWING FORWARD.
C
      DO 8 I=1,NV
      J=2*I-1
      K=J+1
      VXP(1,I,1)=VX(I)
      VXP(1,I,2)=VY(I)
      XO(J)=VX(I)
    8 XO(K)=VY(I)
C
C        XIP= X- VALUES AT THE NIP X-STATIONS AT WHICH OUTPUT NEQD.
C        FIRST VALUE FOR XIP CAN BE ANY NUMBER LARGER THAN ZERO
C        INTEGRATION STARTS AT XIP(1)
C        INTEGRATION STEP SIZE DETERMINED BY DASCRU...H
C
C
C
C
C  NOTES HERE  BY.....HORIZONTAL SEMI-AXIS.
C              AZ.....VERTICAL SEMI-AXIS.
C
C
      CALL ELLSHP (XIP(1),BY,AZ,BYP,AZP)
      SLOC = 0.0
C
C  ......COMPUTE VORTEX PATHS USING DASCRU INTEGRATION SUBROUTINE......
C
      NIP1=NIP-1
      H=XIP(2)-XIP(1)
      DO 20 I=1,NIP1
      I1=I+1
      NV=NVV(I)
      NVN=NVV(I1)
      NV1=NV+1
      N=2*NV
      XIP1=XIP(I+1)
C
      CALL DASCRU(XIP(I),XIP1,H,N,XO,WK,IER,ES)
C
C        IF IER GREATER THAN 32,DASCRU HAD TROUBLE GETTING STEP SIZE.
C        THEN,RATHER THAN CONTINUING WITH THE INTEGRATION, STOP 40
C        REDUCE XIP(2)-XIP(1)AND RERUN
C
      IF (IER.GE.32) GO TO 28
      GO TO 29
   28 WRITE (6,701)
      STOP 40
   29 CONTINUE
C
C
C  .. ADD THE NEW VORTICES AND THEIR POSITIONS FOR THE NEXT X-INTERVAL
C
      IF(NVN.EQ.NV)GO TO 39
      DO 35 K1=NV1,NVN
      JL=2*K1-1
      KK=JL+1
      XO(JL)=VX(K1)
   35 XO(KK)=VY(K1)
   39 CONTINUE
C
C  .. OUTPUT VORTEX POSITIONS AND CONFIGURATION CHARACTERISTICS AT THIS
C     STATION.
C
      CALL ELLSHP (XIP1,BY,AZ,BYP,AZP)
      DO 23 L=1,NVN
      J=2*L-1
      K=J+1
      VXP(I1,L,1)=XO(J)
      VXP(I1,L,2)=XO(K)
   23 CONTINUE
   20 CONTINUE
C
C  ..COMPUTE VELOCITIES AT THE SPECIFIED CONTROL POINTS ON THE BODY
C   INDUCED BY NV EXTERNAL VORTICES AND THEIR IMAGES.
C
C
C  NOTES IF CONTROL POINTS ARE PASSED THROUGH BY MEANS OF A DATA SET
C     I.E. WHEN NCPIN IS NOT EQUAL TO 0.    INDEX NCP IS READ IN FROM
C     THE DATA SET ALSO.
C  NOTES X-COORDINATE OF CONTROL POINT MUST THEN BE TRANSFORMED TO BODY
C     COORDINATE SYSTEM.
C     THIS IS DONE BELOW.
C
C
      IF(NCP.EQ.0)GO TO 99
      DO 69 J11=1,NCP
      CPX = XCP(J11)
```

Figure C-1(uuu)

```
      CPY = YCP(J11)
      CPZ = ZCP(J11)
C
C.. DETERMINE THE X-STATIONS ADJACENT TO THE CONTROL POINT
C
      DO 61 I=1,NIP
      IF (CPX.LT.XIP(I)) GO TO 69
      J=I-1
      IF (I.EQ.1) J=1
      IF(CPX.LE.XIP(I))GO TO 62
   61 CONTINUE
C
C  DETERMINE BY INTERPOLATION THE POSITION IN THE CROSSFLOW PLANE OF ALL
C     VORTICES AT EACH STATION CPX.
C
   62 K=J+1
      X1=XIP(J)
      X2=XIP(K)
      WT1=(X2-CPX)/(X2-X1)
      WT2=(CPX-X1)/(X2-X1)
      NV=NVV(J)
      CALL ELLSHP (CPX,BY,AZ,BYP,AZP)
      DO 63 I=1,NV
      VX(I)=WT1*VXP(J,I,1)+WT2*VXP(K,I,1)
      VY(I)=WT1*VXP(J,I,2)+WT2*VXP(K,I,2)
   63 CONTINUE
C
      THETP=ATAN2(CPZ,CPY)
      SINTH=SIN(THETP)
      COSTH=COS(THETP)
      RCPT=SQRT(CPY*CPY+CPZ*CPZ)
      RBODY=SQRT(1.0/((SINTH/AZ)**2+(COSTH/BY)**2))
      IF (RCPT.LE.RBODY) GO TO 64
      GO TO 65
   64 RBODYP=1.01*RBODY
      CPY=RBODYP*COSTH
      CPZ=RBODYP*SINTH
   65 CONTINUE
C
C     ROUTINE VVELS CALCULATES VELOCITIES INDUCED BY EXTERNAL
C     VORTICES AROUND ELLIPTICAL CROSS SECTION BODY BY SETTING S=RB IN
C     THE ARGUMENT LIST.
C
      CALL VVELS(NV,CPY,CPZ,VX,VY,G,BY,AZ,BY,V,W,VRTMAX)
      VCP(J11) = V
      WCP(J11) = W
   69 CONTINUE
   99 RETURN
      END



      SUBROUTINE VVELS(NV,YY,ZZ,VX,VY,G,AB,BB,V,W,VRTMAX)
C
C     VERSION( DEMON2
C
C     THIS SUBROUTINE COMPUTES PERTURBATION VELOCITY COMPONENTS DUE TO
C     NV EXTERNAL VORTICES AND THEIR IMAGES INSIDE A BODY WITH
C     ELLIPTICAL CROSS SECTION. THEY ARE ADDED TO V AND W IN THE
C     ARGUMENT LIST.
C     A DESCRIPTION OF THE METHODS AND EQUATIONS PROGRAMMED HERE
C     ARE GIVEN IN SECTION 5.1 AND 5.2 OF APPENDIX I OF NASA CR-3122
C
C
      DIMENSION VX(1),VY(1),G(1)
C
      COMMON/COM1/AZ,B2,R2
      COMMON/COM2/SIG2,M2
C
      COMPLEX TO,V1,DSDT,Z,DSDZ,SI,SIB,SO,TAU,CI,VEL
C
      EXTERNAL Z,DSDZ
      PI=3.14159265
      TLC=0.001
      CI=CMPLX(0.0,1.0)
      A2=AB*AB
      B2=BB*BB
      APB=AB+BB
      APB2=APB*APB
      R=0.5*APB
      M2=APB2
      R2=R*R
      SIG2=R2
      TO=CMPLX(YY,ZZ)
      SO=Z(TO)
      V1=CMPLX(0.0,0.0)
      DSDT=DSDZ(SO)
C
C  LOOP OVER THE NUMBER OF VORTICES,NV
C
      DO 1 I=1,NV
      TAU=CMPLX(VX(I),VY(I))
      SI=Z(TAU)
      SIB=CONJG(SI)
      D=CABS(SI-SO)
      IF(D.LE.TLC) GO TO 2
      V1=V1-G(I)/(SO-SI)
    2 CONTINUE
      D=CABS(SO-R2/SIB)
      IF(D.LE.TLC) GO TO 1
      V1=V1+G(I)/(SO-R2/SIB)
    1 CONTINUE
C
      VEL=0.5*CI*V1*DSDT/PI
      V=REAL(VEL)+V
      W=-AIMAG(VEL)+W
      AV=ABS(V)
      AW=ABS(W)
      IF(V.GT.0.0.AND.AV.GE.VRTMAX) V=VRTMAX
      IF(V.LT.0.0.AND.AV.GE.VRTMAX)V=-VRTMAX
      IF(W.GT.0.0.AND.AW.GE.VRTMAX) W=VRTMAX
      IF(W.LT.0.0.AND.AW.GE.VRTMAX)W=-VRTMAX
      RETURN
      END


      SUBROUTINE VWAVG (V,W,X,NRING,IROW)
C
C  ROUTINE TO AVERAGE VELOCITY COMPONENTS RING BY RING
C
      DIMENSION V(1),W(1),X(1),IROW(50)
      COMMON /WDY1  / VAV(50),WAV(50),XAV(50),NR
C
      NR = NRING
      I = 0
      DO 120 K=1,NRING
      JR = IROW(K)
      SV = 0.0
      SW = 0.0
      SX = 0.0
      SJR = FLOAT(JR)
      DO 100 J=1,JR
      I = I+1
      SV = SV+V(I)
      SW = SW+W(I)
      SX = SX+X(I)
  100 CONTINUE
      VAV(K) = SV/SJR
      WAV(K) = SW/SJR
      XAV(K) = SX/SJR
  120 CONTINUE
      RETURN
      END
```

Figure C-1(vvv)

```
      SUBROUTINE VXYZ
C
C  ROUTINE TO COMPUTE THE VELOCITY COMPONENTS DUE TO FREE STREAM
C  FROM TRANSLATION OF STORE BODY
C
      DIMENSION DUMMY(12)
      COMMON /RVELFS/ VX,VY,VZ,VRATS,RVX,RVY,RVZ
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     ,CDC,XMOM,ESTRMX,DX,DELA,VSTORE,ESTLOC
      COMMON /FLOW  / ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /IFORCE/ NDAMP,NEJSTR,NEMP,NGAM,NSEG ,NHSEGO,NROLL
      COMMON /PARAM / XMACH,ALPHA,BET2,ALPHAC,PHIR,EM
     1  ,SINAC,COSAC,SPHI,CPHI,SINA,SINB
      DATA  RTOD/57.29577951/
C
C  COMPUTE DIRECTION COSINES
C
      IF (NGAM.GT.0) GO TO 100
      CALL DIRCOS(VAR,DC)
      GO TO 110
100   TA = VINF*COS(ALFACR)+VAR(1)
      DUMMY(10)=VAR(10)-ATAN(VAR(2)/TA)
      DUMMY(11)=VAR(11)+ATAN(VAR(3)/TA)
      DUMMY(12)=VAR(12)
      CALL DIRCOS(DUMMY,DC)
110   CONTINUE
C
      VXI = VINF*COS(ALFACR)
      VETA = 0.0
      VZETA = VINF*SIN(ALFACR)
      IF (NGAM.EQ.1) GO TO 120
      VXI = VXI+VAR(1)
      VETA = VAR(2)
      VZETA = VZETA+VAR(3)
120   VSTORE = SQRT(VXI*VXI+VETA*VETA+VZETA*VZETA)
      CALL INTOST (VXI,VETA,VZETA,VX,VY,VZ,DC)
      VRATS = VINF/VSTORE
      RVX = VX/VSTORE
      RVY = VY/VSTORE
      RVZ = VZ/VSTORE
C
C  CALCULATE STORE FREE STREAM VELOCITY
C
      SINA = RVZ
      SINB = RVY
      SIN2 = SINA*SINA+SINB*SINB
      SINAC = SQRT(SIN2)
      COSAC = SQRT(1.-SIN2)
      ALPHA = ASIN(SINA)*RTOD
      BET2  = ASIN(SINB)*RTOD
      ALPHAC= ASIN(SINAC)*RTOD
      PHIR = 0.0
      IF (SINAC.NE.0.) PHIR=ASIN(SINB/SINAC)*RTOD
      RETURN
      END


      SUBROUTINE XVSR (RSHK,XSHK,NSHK,RS,XS,ALP)
C
C  ROUTINE TO PERFORM A SINGLE SEARCH OF TABLE DATA AND PERFORM AN
C  INTERPOLATION BETWEEN TABLE VALUES TO FIND X VERSUS R.
C  ALP = ANGLE OF ATTACK EFFECT SEEN BY SHOCK TRAVERSE
C        ALP = (ALPHAC-SIB)(1-EALPHA)COS(PHIS(K)-PHIR)
C
      DIMENSION RSHK(NSHK),XSHK(NSHK)
      FX(X,R) =  X*CALP+R*SALP
      FR(X,R) = -X*SALP+R*CALP
      SALP = SIN(ALP)
      CALP = COS(ALP)
      DO 100 I=2,NSHK
      IF (RS.LE.FR(XSHK(I),RSHK(I))) GO TO 120
100   CONTINUE
      I = NSHK
120   XA  = FX(XSHK(I),RSHK(I))
      RA  = FR(XSHK(I),RSHK(I))
      XA1 = FX(XSHK(I-1),RSHK(I-1))
      RA1 = FR(XSHK(I-1),RSHK(I-1))
      XS = XA+(XA1-XA)*(RS-RA)/(RA1-RA)
      RETURN
      END


      SUBROUTINE XVSRT (XS,YS,ZS,NSHK,PHIS,NSHOCK,XSHK,RSHK,ALP,PHIR
     *   ,LSYM)
C
C  ROUTINE TO INTERPOLATE FOR X-LOCATION OF MODIFIED SHOCK SHAPE
C  VERSUS RADIAL DISTANCE, RS(YS,ZS) AROUND NON-CIRCULAR BODY AND
C  ANGULAR LOCATION OF TRAVERSES, PHIS.
C  ALP = ANGLE SHOCK SHAPE IS ROTATED RELATIVE TO UNROLLED BODY
C           ALP = (ALFACR-SIB(R))(1-EALPHA)
C  PHIR = STORE ROLL ANGLE, MEASURED COUNTERCLOCKWISE FROM Z DOWN
C  LSYM = TRUE.  SHAPE COMPUTED FOR ALPHA=0 FROM PHIS=0 TO 90.
C                SYMMETRIC  TOP TO BOTTOM AND RIGHT TO LEFT.
C                ALPHA CORRECTION SYMMETRIC RIGHT TO LEFT ONLY.
C         FALSE. RIGHT/LEFT SYMMETRY. ALPHA CORRECTION IMPLICITE
C
      LOGICAL LSYM
      DIMENSION NSHK(10),PHIS(10),XSHK(100),RSHK(100)
      COMMON /FLOW/ ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ
      COMMON /CONSTS/ PI,DTOR
      RS = SQRT(YS*YS+ZS*ZS)
      XS = 0.0
      IF (NSHOCK.LE.0) GO TO 200
      IF (RS.LE.0.) GO TO 999
      THTS = ATAN2(YS,-ZS)
      IF (THTS.LT.0.) THTS=THTS+2.*PI
      THTO = THTS
      IF (THTS.GT.PI) THTS=ABS(THTS-2.*PI)
      IF (LSYM .AND. THTS.GT.0.5*PI) THTS=ABS(THTS-PI)
      PHI = THTO-THTS+PHIR*DTOR
      IF (NSHOCK.EQ.1) GO TO 150
      I2 = 1
C
C  FIND TRAVERSE ANGLES BETWEEN WHICH POINT LIES
C
      THTD = THTS/DTOR
      DO 100 J=2,NSHOCK
      JM1 = J-1
      I1 = I2
      I2 = I1+NSHK(JM1)
      IF (THTD.LE.PHIS(J)) GO TO 110
100   CONTINUE
      J = NSHOCK
C
C  CALCULATE X AT RS FOR PHIS ON EITHER SIDE OF TRAVERSE
C
110   CONTINUE
      ALF = 0.0
      IF (LSYM) ALF=ALP*COS(PHI+PHIS(JM1)*DTOR)
      CALL XVSR (RSHK(I1),XSHK(I1),NSHK(JM1),RS,XSJM1,ALF)
      IF (LSYM) ALF=ALP*COS(PHI+PHIS(J)*DTOR)
      CALL XVSR (RSHK(I2),XSHK(I2),NSHK(J),RS,XSJ,ALF)
      XS = XSJM1+(THTD-PHIS(JM1))*(XSJ-XSJM1)/(PHIS(J)-PHIS(JM1))
      GO TO 999
C
C  SINGLE SHOCK SHAPE TRAVERSE EXISTS -----------------------------------
C
150   ALF = ALP*COS(THTO-PHIR*DTOR)
      CALL XVSR (RSHK,XSHK,NSHK,RS,XS,ALF)
      GO TO 999
C
C  LINEAR THEORY SHOCK SHAPE --------------------------------------------
```

Figure C-1(www)

```
C
200   TANA = TAN(ALP*COS(PHIR*DTOR))
      XS = XSHK(1)+HS*(HETA+TANA)/(1.-BETA*TANA)
999   CONTINUE
      RETURN
      END
```

```
      COMPLEX FUNCTION Z(CT)
C     VERSION( DEMON2
C
C
C     THIS FUNCTION SUBROUTINE CALCULATES THE SIGMA VALUE IN THE
C     TRANSFORMED (CIRCLE) PLANE FOR GIVEN TAU IN THE PHYSICAL PLANE
C     FOR AN ELLIPTICAL BODY WITH WINGS
C     A DESCRIPTION OF THE METHODS AND EQUATIONS PROGRAMMED HERE
C     ARE GIVEN IN SECTION 5.1  OF APPENDIX I OF NASA CR-3122
C     ROUTINE PERFORMS CONFORMAL MAPPING.
C
      COMMON/COM2/SIG2,H2
      COMMON/COM3/ZR,ZI
      COMMON/COM4/G2,G1
      COMMON/COM6/W2,W
C
C
      COMPLEX W,G1,G2,CT,W2,DBLU
C
      EXTERNAL DBLU
C
      W=DBLU(CT)
      G1=W+SIG2/W
      G2=G1*G1-H2
      Y=AIMAG(G2)
      AY=1.0
      IF(Y.LT.0.0) AY=-1.0
      YZ=AIMAG(G1)
      AYZ=1.0
      IF(YZ.LT.0.0)AYZ=-1.0
      G2=CSQRT(G2)*AY*AYZ
      IF((ABS(YZ).LE.0.0).AND.(REAL(G1).LT.0.0)) G2=CMPLX(-REAL(G2),
     1     AIMAG(G2))
      Z=0.5*(G1+G2)
      IF((ABS(ZI).NE.0.0).AND.(ABS(ZR).NE.0.0)) Z=CMPLX(ZR*ABS(REAL(Z)),
     1ZI*ABS(AIMAG(Z)))
      RETURN
      END
```

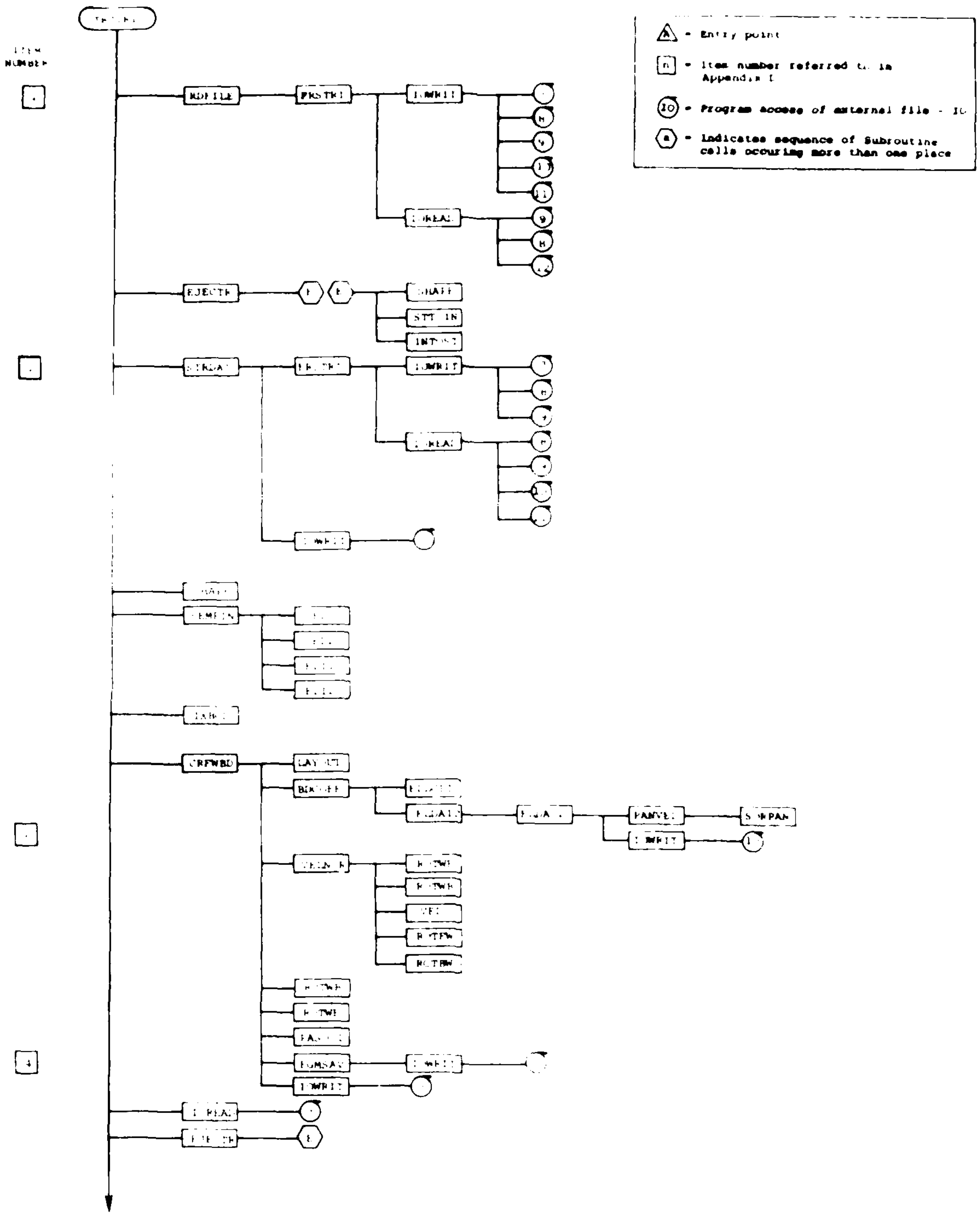
```
      SUBROUTINE ZIMSVL(NM,XFP,RFP,CTHET,STHET,BETAAL,BETASL,USD,VSD,
     *   WSD,FAC,CTHER,STHER)
C
C     THIS SUBROUTINE CALCULATES THE IMAGE STORE VELOCITIES
C              AT FIN CONTROL POINTS
C
C
      COMMON /SORDUB/ BETAA,SRS(150),VRATSA,BETAS,BSS,DSW(150),
     1 DSV(150),XS(150),AD(150),BETAAI,BETASI,BSSI
      COMMON /TRMSSD/ ARG,ROOT,ACOSH
C
C     SOURCE VELOCITIES
C
      SAX=0.0
      SRAD=0.0
      DO 32 NN=1,NM
      BARR=RFP*BETAAL
      ARG=(XFP-XS(NN))/BARR
      IF (ARG.LE.1.0) GO TO 33
      CALL SDTRMS
      SS=SRS(NN)
      SAX=SAX-SS*ACOSH
   32 SRAD=SRAD+SS*BETAAL*ROOT
   33 CONTINUE
      SAX=SAX/VRATSA
      SRAD=SRAD/VRATSA
C
C     DOUBLET VELOCITIES
C
      DWAX=0.0
      DVAX=0.0
      DWRAD=0.0
      DVRAD=0.0
      DWTAN=0.0
      DVTAN=0.0
      IF (FAC.LE.0.0) GO TO 41
      DO 40 NN=1,NM
      BSRR=RFP*BETASL
      ARG=(XFP-AD(NN))/BSRR
      IF (ARG.LE.1.0) GO TO 40
      DW=DSW(NN)
      DV=DSV(NN)
      CALL SDTRMS
      TA=0.5*BETASL*BETASL
      TB=BETASL*ROOT
      DWAX=DWAX+DW*TB*CTHET
      DVAX=DVAX+DV*TB*STHET
      TC=ARG*ROOT
      TD=TA*(TC-ACOSH)
      TE=TA*(TC+ACOSH)
      DWRAD=DWRAD-DW*TE*CTHET
      DVRAD=DVRAD-DV*TE*STHET
      DWTAN=DWTAN-DW*TD*STHET
   40 DVTAN=DVTAN+DV*TD*CTHET
   41 CONTINUE
C
C     SUM UP IMAGE VELOCITIES
C
      USD=SAX+FAC*(DWAX+DVAX)
      RADSD=SRAD+FAC*(DWRAD+DVRAD)
      TANSD=FAC*(DWTAN+DVTAN)
      VSD=+RADSD*STHER+TANSD*CTHER
      WSD=-RADSD*CTHER+TANSD*STHER
      RETURN
      END
```

```
      SUBROUTINE ZIMAGE(XFP,YFP,ZFP,UIM,VIM,WIM)
C
C     THIS SUBROUTINE CALCULATE THE SOURCE AND THE DOUBLET VELOCITIES AT
C     THE SEPERATED STORE FINS DUE TO THE IMAGE STORES
C
      COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3)
     *     ,CDC,XMOM,ESTRMA,DX,DFLX,VSTORE,ESTLGC
      COMMON /NEWFOR/ NTHETA,DTHETA,THETAD(37),STHETA(37),CTHETA(37),
     1REFL,SREF,NCONPT
      COMMON /SORDUB/ BETAA,SRS(150),VRATSA,BETAS,BSS,DSW(150),
     1 DSV(150),XS(150),XD(150),BETAAI,BETASI,BSSI
      COMMON/IFORCE/NCAMP,NEJSTR,NEMP,NGAM,NHSEG,NHSEGO,NROLL
      COMMON/XSHOLD/ FXSHLD,HXSHLD,SXSHLD(7)
      COMMON /STORIC/ RTNOSC,RSSC
      COMMON/STORIF/ IMFSTR,RTNOSF,RSS,YBN,ZBN,SFAC
      COMMON /STORIW/ IMSTOW,HTNOSE,XWN,YWN,ZWN,IMAGE
      COMMON /STORSD/ NXBODY,XHD(151),RAD(151),NPOLY,XEND(7),COEF(7,7),
     1 XRC(150),RHC(150),DRDXRC(150),NSORCE,DRDXBD(151)
      COMMON /STRRET/ RTAL,BTSL
      COMMON /STRATC/ RTALC,BTSLC
      COMMON /STRATF/ RTALF,BTSLF
C
C     LOCATE WING IMAGE STORE IN STORE COORDINATE SYSTEM
C
      UIM=0.
      VIM=0.
      WIM=0.
```

Figure C-1(xxx)

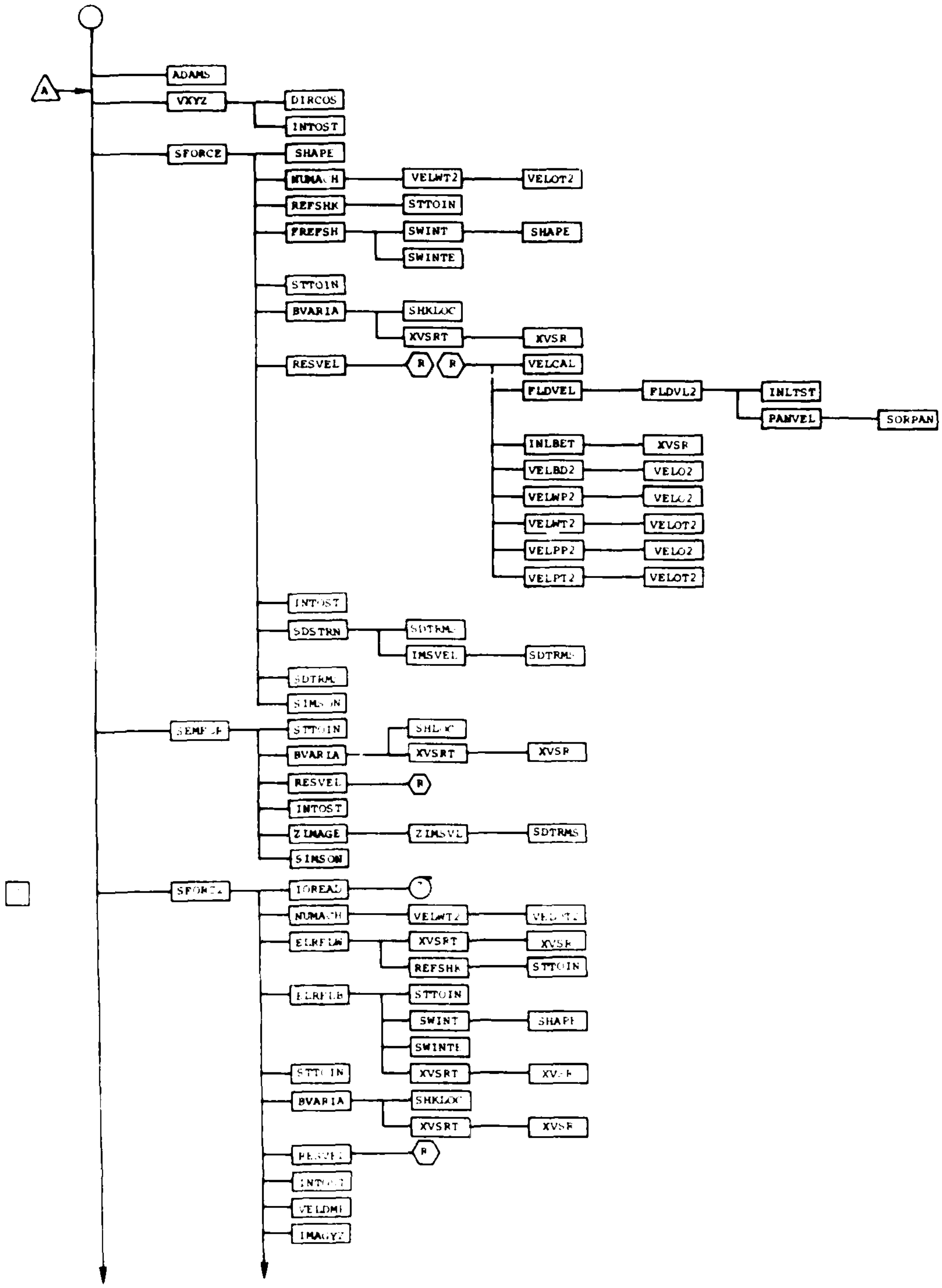
```
      NM=NSORCE
      FMACHA=SQRT(BETAA*BETAA+1.0)
      FMACHS=SQRT(BETAS*BETAS+1.0)
      XSHOL=SXSHLD(NEJSTR)
      IF (IMSTOR.EQ.0) GO TO 1
      CPHI=COS(VAR(12))
      SPHI=SIN(VAR(12))
      ETA=YFP*CPHI-ZFP*SPHI
      ZTA=YFP*SPHI+ZFP*CPHI
      TA=ZWN+ZWN+ZTA
      RFP=SQRT(ETA*ETA+TA*TA)
      PHIFP=ATAN(ETA/TA)
      XSHKI=RFP*BTNOSE
      XINAI=RFP*BETAA+XSHOL
      XINSI=RFP*BETAS+XSHOL
      FMSHI=SQRT(BTNOSE*BTNOSE+1.0)
      STHET=SIN(-VAR(12)+PHIFP)
      CTHET=COS(-VAR(12)+PHIFP)
      STHER=SIN(VAR(12)+PHIFP)
      CTHER=COS(VAR(12)+PHIFP)
C
C     LOCATE FUSELAGE IMAGE STORE IN STORE COORDINATE SYSTEM
C
    1 IF (IMFSTR.EQ.0) GO TO 2
      THONE=ATAN(-YBN/ZBN)-VAR(12)
      CTHONE=COS(THONE)
      STHONE=SIN(THONE)
      ETA=YFP*CTHONE+ZFP*STHONE
      ZTA=-YFP*STHONE+ZFP*CTHONE
      TA=RSS+ZTA
      RFPF=SQRT(ETA*ETA+TA*TA)
      PHIFPF=ATAN(ETA/TA)
      THETF=THONE+PHIFPF
      STHETF=SIN(THETF)
      CTHETF=COS(THETF)
      THERF=PHIFPF-THONE
      STHERF=SIN(THERF)
      CTHERF=COS(THERF)
      TA=RSSC+ZTA
      RFPC=SQRT(ETA*ETA+TA*TA)
      PHIFPC=ATAN(ETA/TA)
      THETC=THONE+PHIFPC
      STHETC=SIN(THETC)
      CTHETC=COS(THETC)
      THERC=PHIFPC-THONE
      STHERC=SIN(THERC)
      CTHERC=COS(THERC)
      XSHKIF=RFPF*BTNOSF
      XINAIF=RFPF*BETAA+XSHOL
      XINSIF=RFPF*BETAS+XSHOL
      FMSHIF=SQRT(BTNOSF*BTNOSF+1.0)
    2 CONTINUE
C
C     CALCULATE IMAGE STORE VELOCITIES
C
C
C     WING IMAGE STORE
C
      IF (IMSTOR.EQ.0) GO TO 10
      BTAL=BETAA
      BTSL=BETAS
      IF (XFP.LT.XSHKI) GO TO 11
      IF (XFP.GT.XSHKI) GO TO 12
      BTAL=BTNOSE
      BTSL=BTNOSE
      GO TO 11
   12 IF (XFP.GE.XINSI) GO TO 11
      FMLS=FMSHI+(FMACHS-FMSHI)*(XFP-XSHKI)/(XINSI-XSHKI)
      BTSL=SQRT(FMLS*FMLS-1.0)
      IF (XFP.GE.XINAI) GO TO 11
      FMLA=FMSHI+(FMACHA-FMSHI)*(XFP-XSHKI)/(XINAI-XSHKI)
      BTAL=SQRT(FMLA*FMLA-1.0)
   11 CONTINUE
      FAC=1.0
      CALL ZIMSVL(NM,XFP,RFP,CTHET,STHET,BTAL,BTSL,USD,VSD,WSD,FAC,
     *CTHER,STHER)
      UIM=USD
      VIM=VSD
      WIM=WSD
C
C     FUSELAGE IMAGE STORE
C
   10 IF(IMFSTR.EQ.0) RETURN
      BTALF=BETAA
      BTSLF=BETAS
      IF (XFP.LT.XSHKIF) GO TO 13
      IF (XFP.GT.XSHKIF) GO TO 14
      BTALF=BTNOSF
      BTSLF=BTNOSF
      GO TO 13
   14 IF (XFP.GE.XINSIF) GO TO 13
      FMLS=FMSHIF+(FMACHS-FMSHIF)*(XFP-XSHKIF)/(XINSIF-XSHKIF)
      BTSLF=SQRT(FMLS*FMLS-1.0)
      IF (XFP.GE.XINAIF) GO TO 13
      FMLA=FMSHIF+(FMACHA-FMSHIF)*(XFP-XSHKIF)/(XINAIF-XSHKIF)
      BTALF=SQRT(FMLA*FMLA-1.0)
   13 BTALC=BTALF*RSS/RSSC
      BTSLC=BTSLF*RSS/RSSC
      CALL ZIMSVL(NM,XFP,RFPF,CTHETF,STHETF,BTALF,BTSLF,USD,VSD,WSD,
     *SFAC,CTHERF,STHERF)
      UIM=UIM+USD
      VIM=VIM+VSD
      WIM=WIM+WSD
      FAC=0.0
      CALL ZIMSVL(NM,XFP,RFPC,CTHETC,STHETC,BTALC,BTSLC,USD,VSD,WSD,
     *FAC,CTHERC,STHERC)
      UIM=UIM-USD
      VIM=VIM-VSD
      WIM=WIM-WSD
      RETURN
      END
```

Figure C-1(yyy)



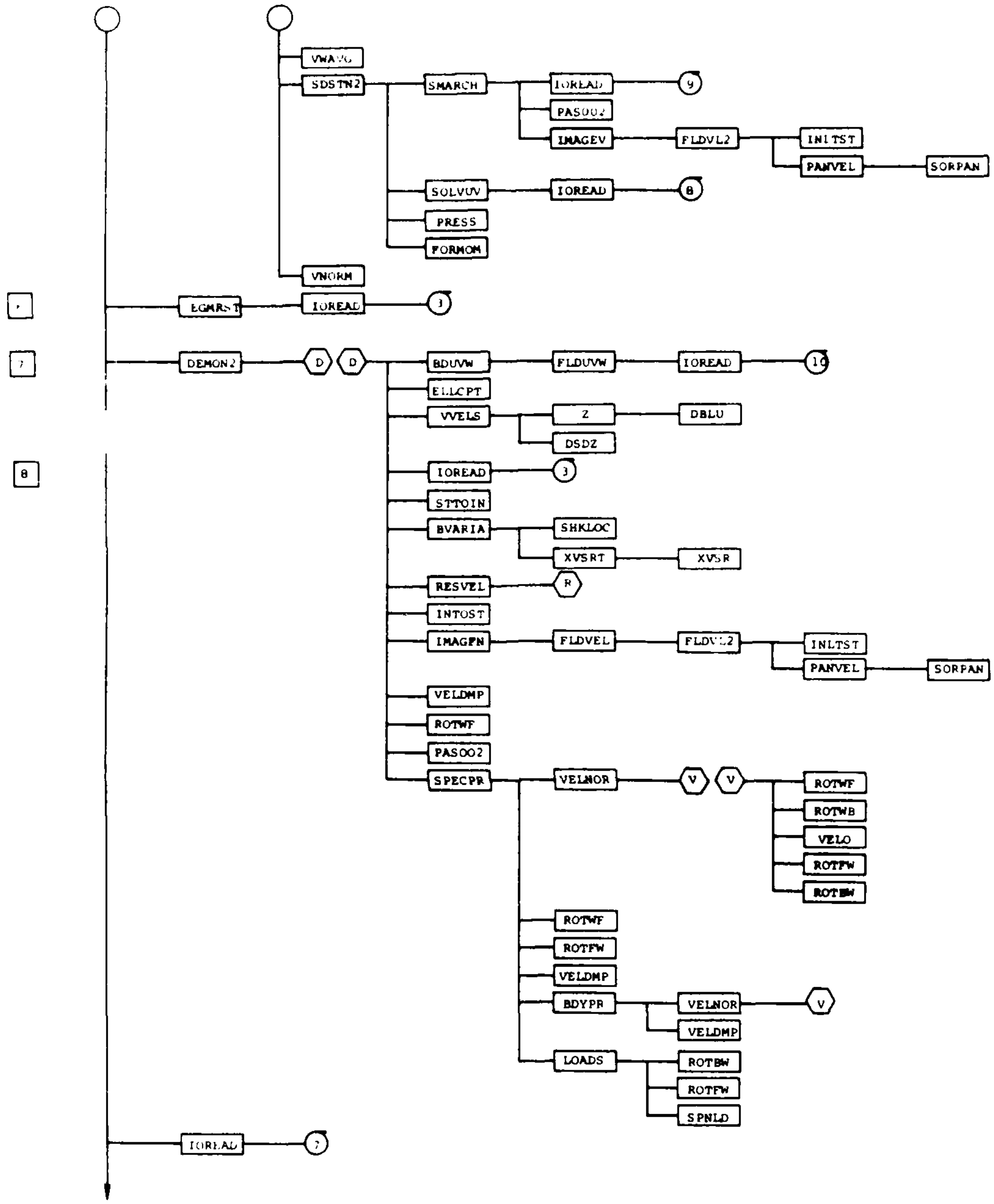
(a)

Figure C-2.- General flow chart of subroutine calls in Program II.



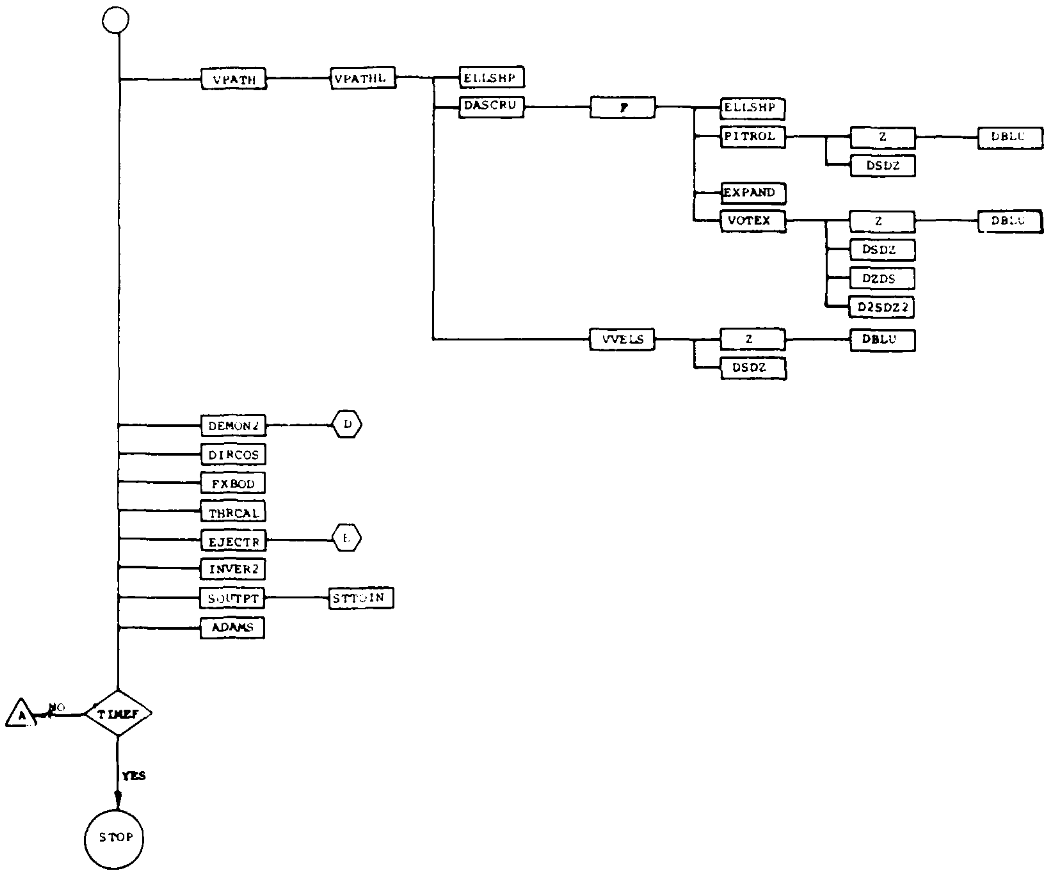
(b)

Figure C-2.- Continued.



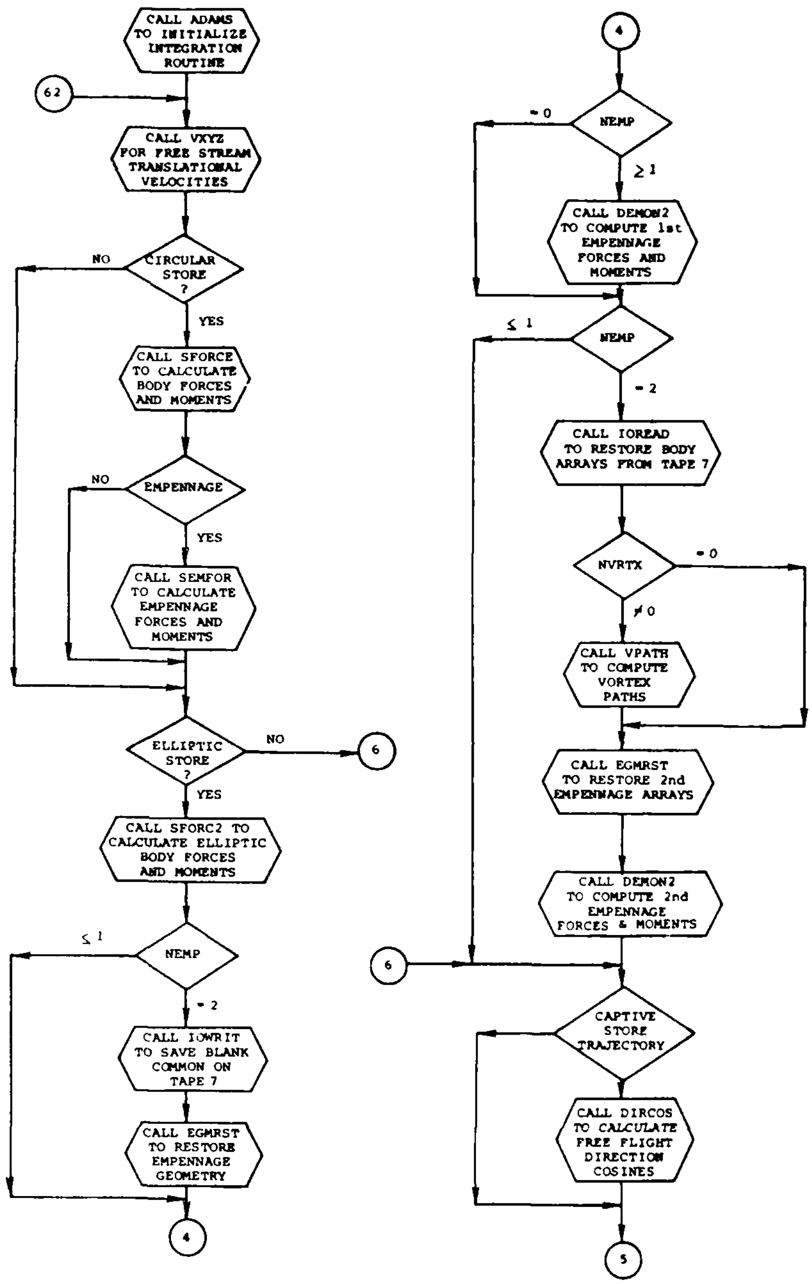
(c)

Figure C-2.- Continued.



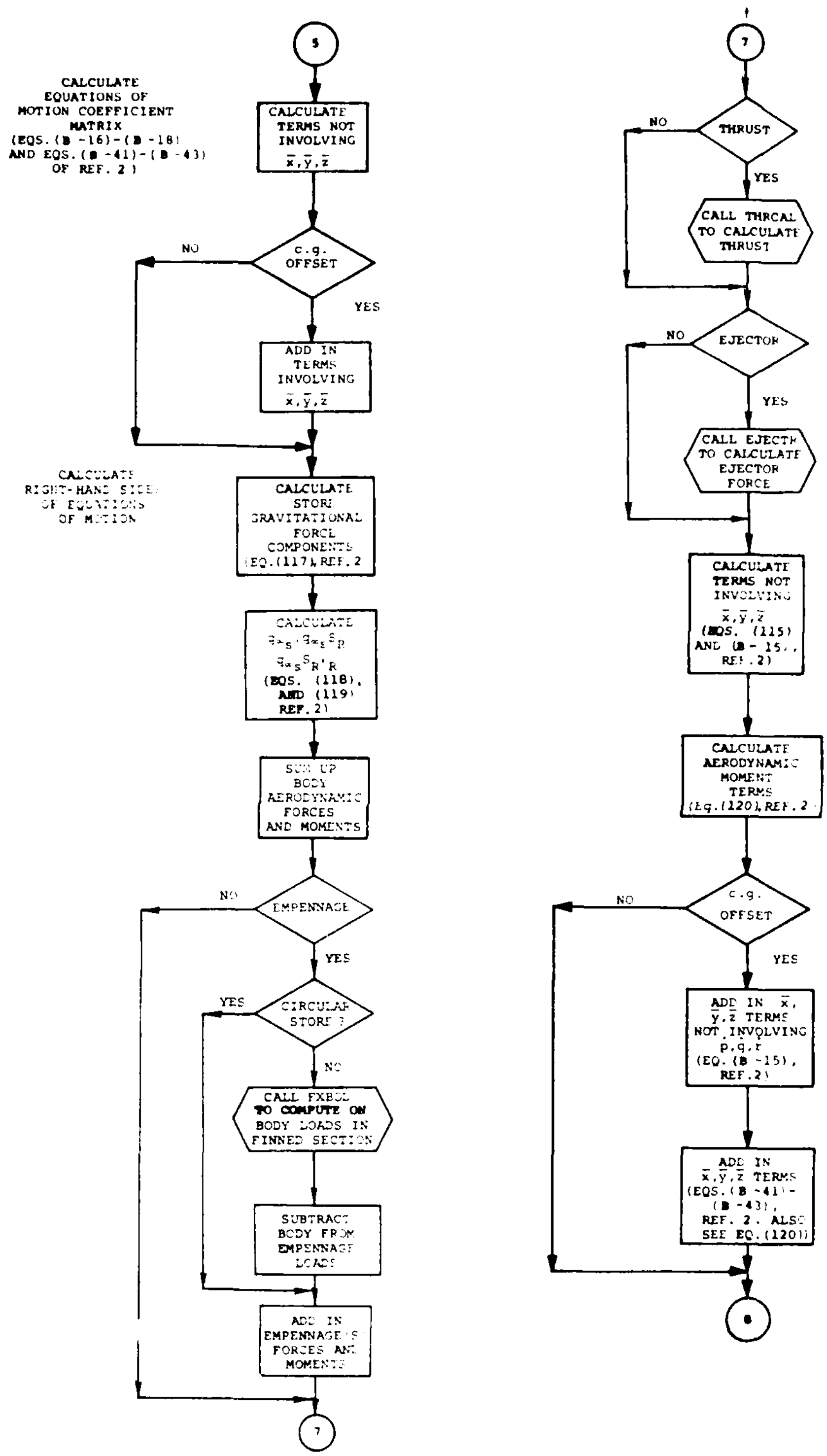
(d)

Figure C-2.- Concluded.



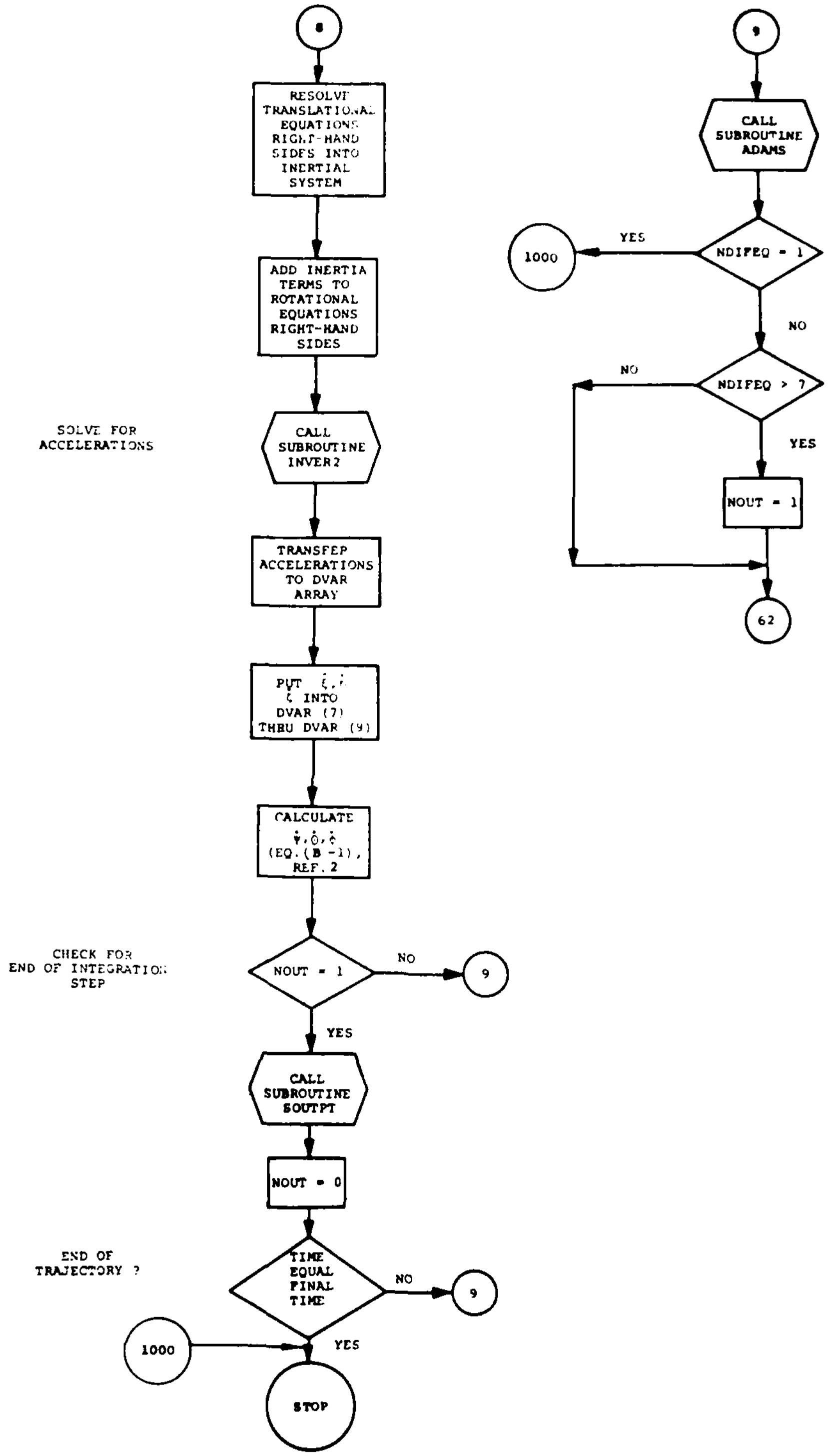
(a)

Figure C-3.- Flow chart of integration loop of main program, TRJTRY.



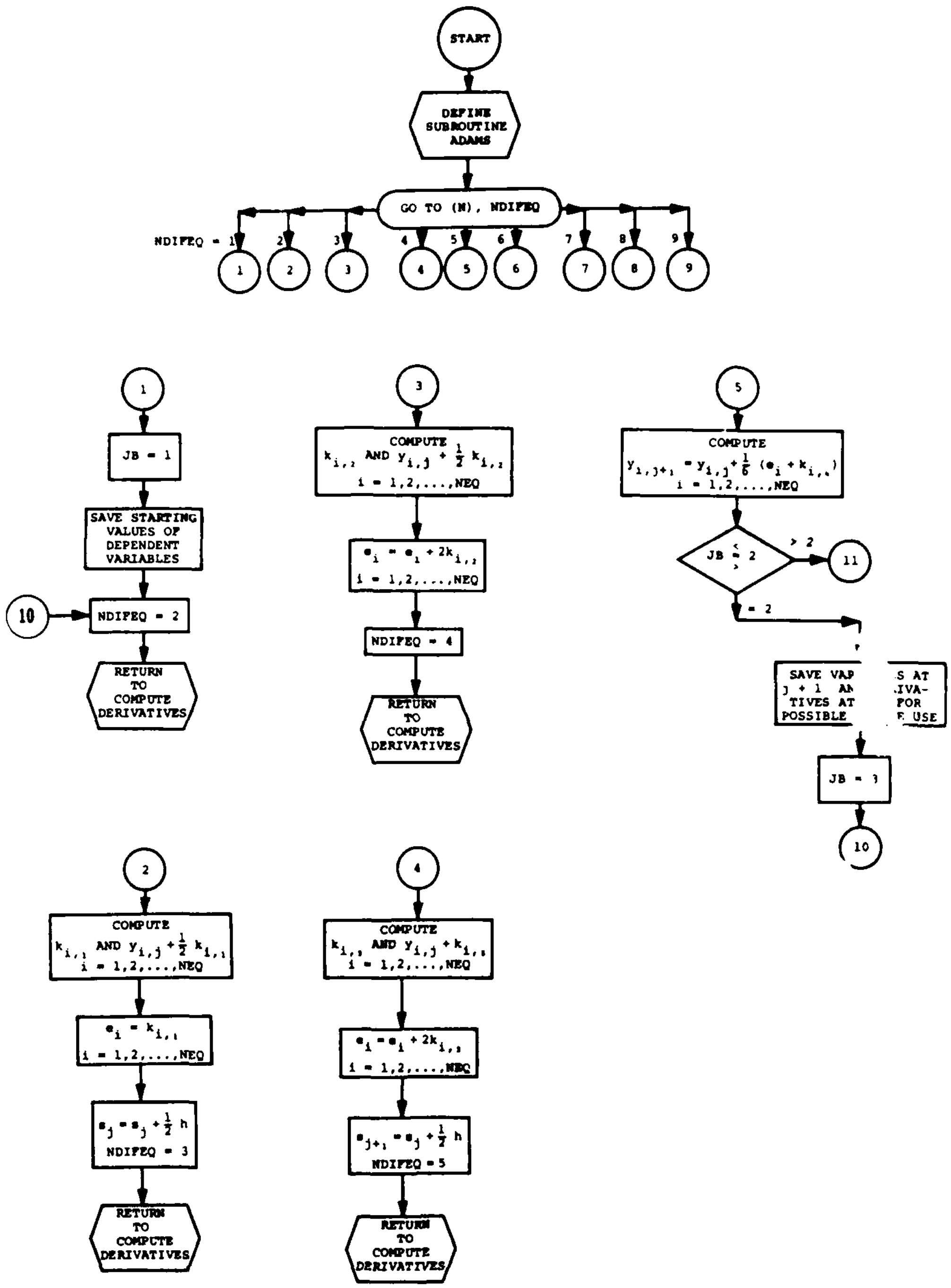
(b)

Figure C-3.- Continued.



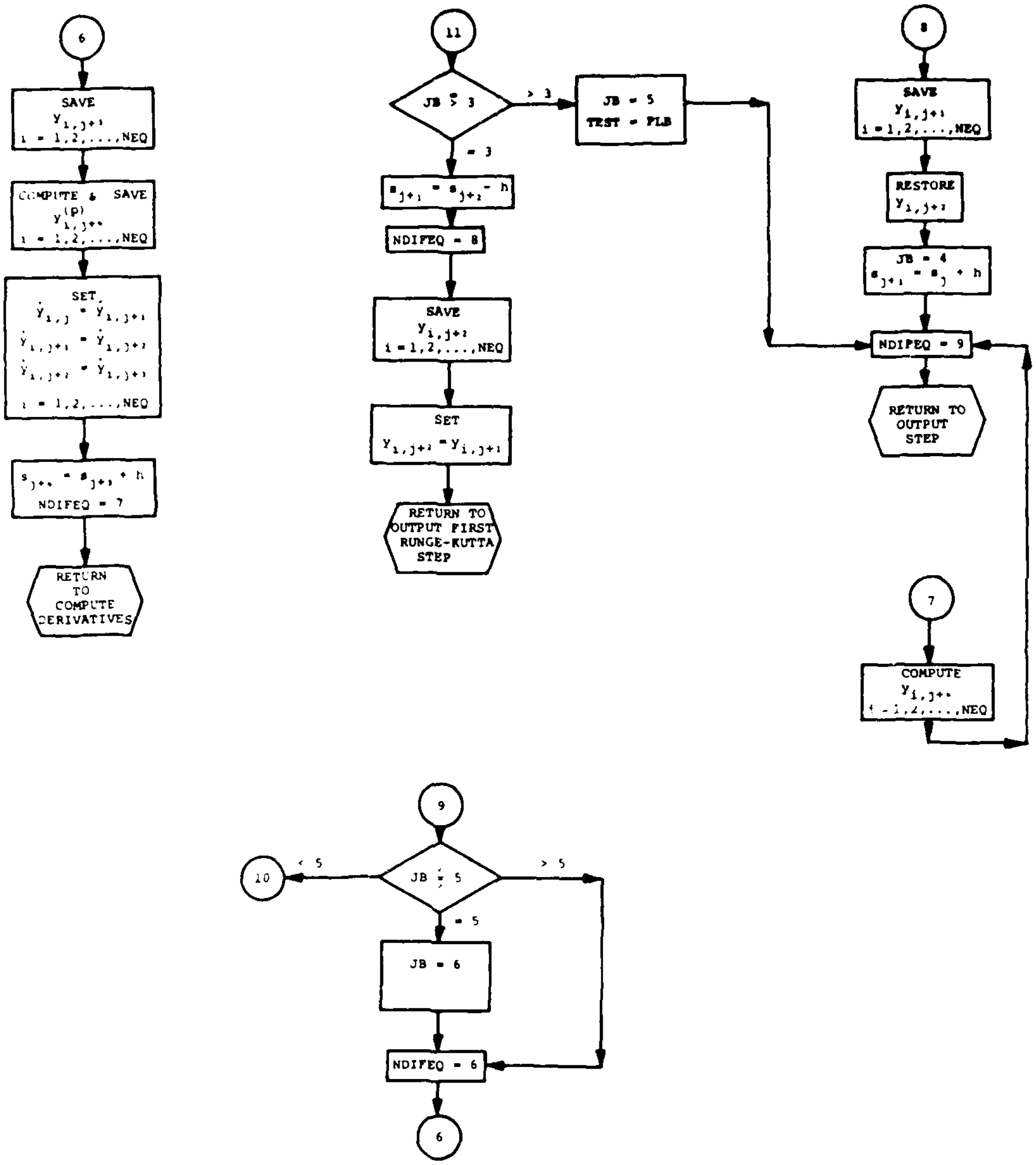
(c)

Figure C-3.- Concluded.



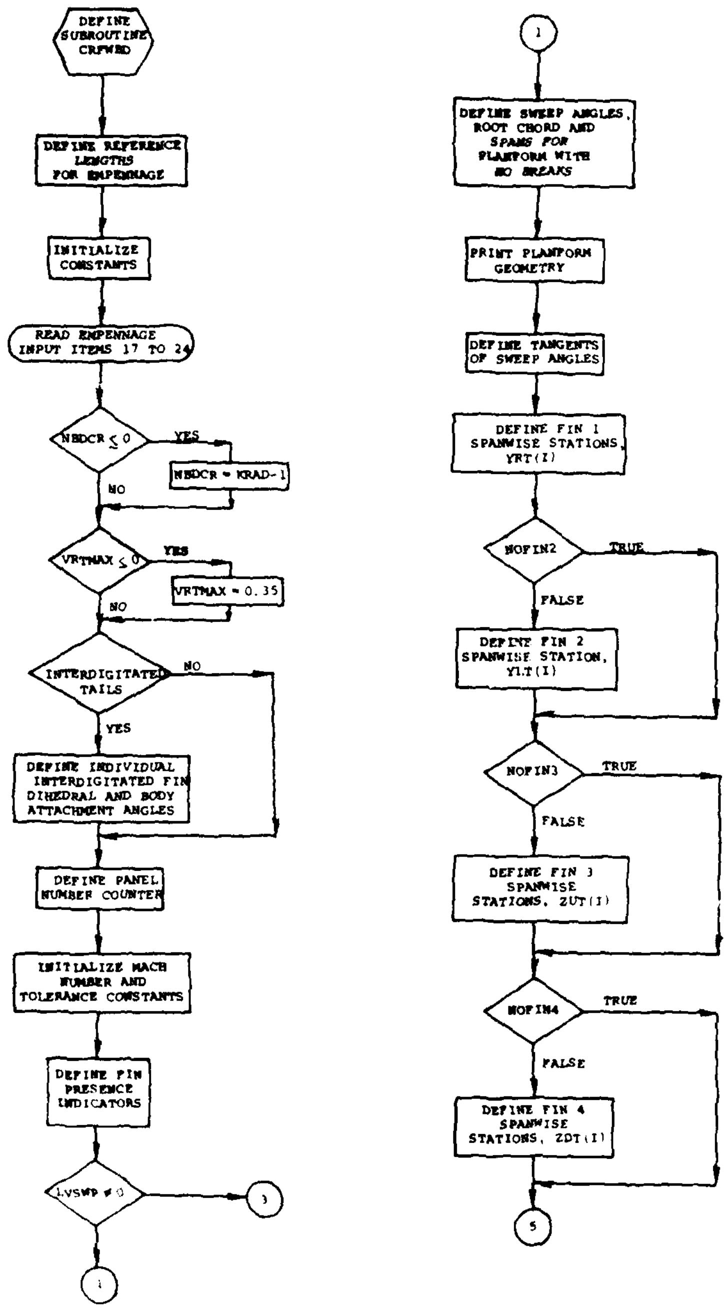
(a)

Figure C-4.- Flow chart of subroutine ADAMS.



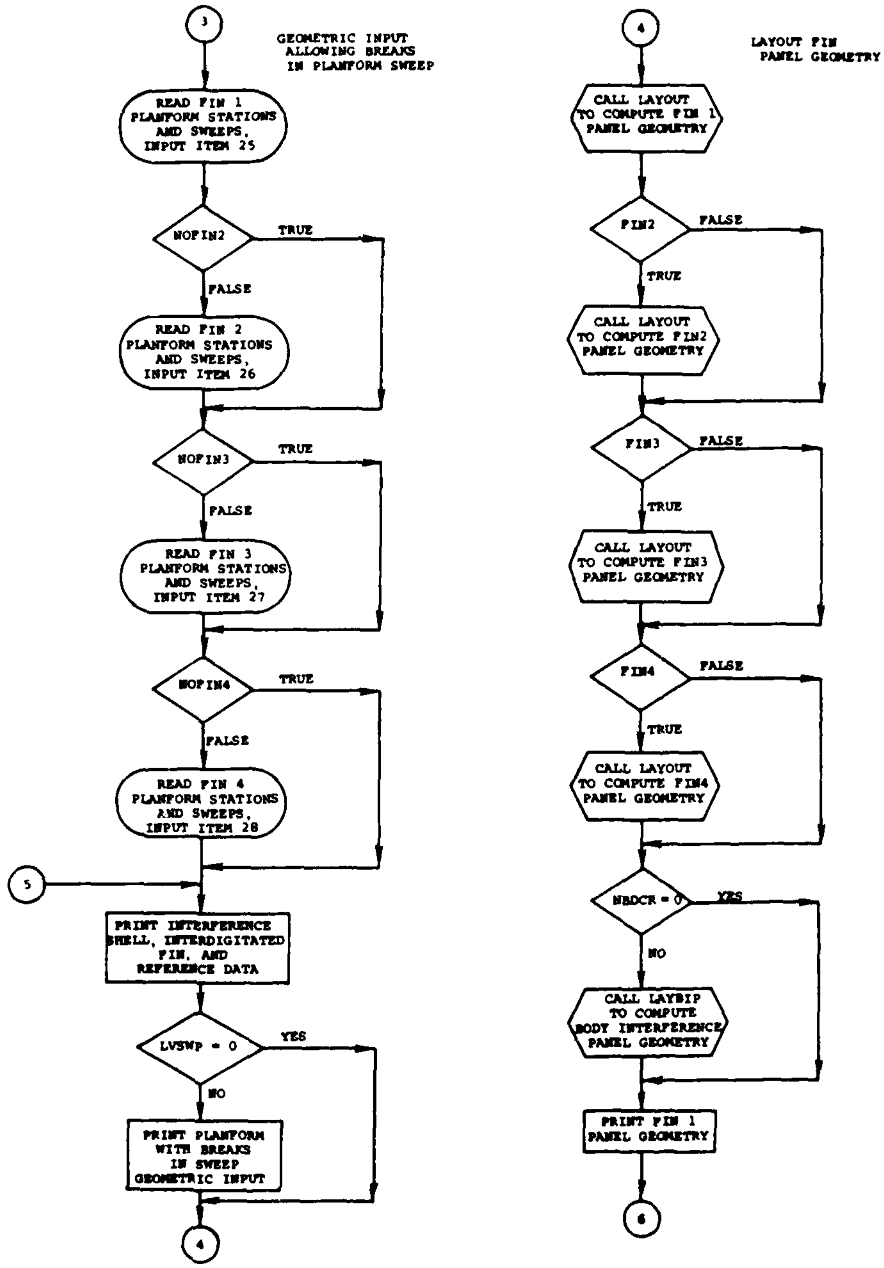
(b)

Figure C-4.- Concluded.



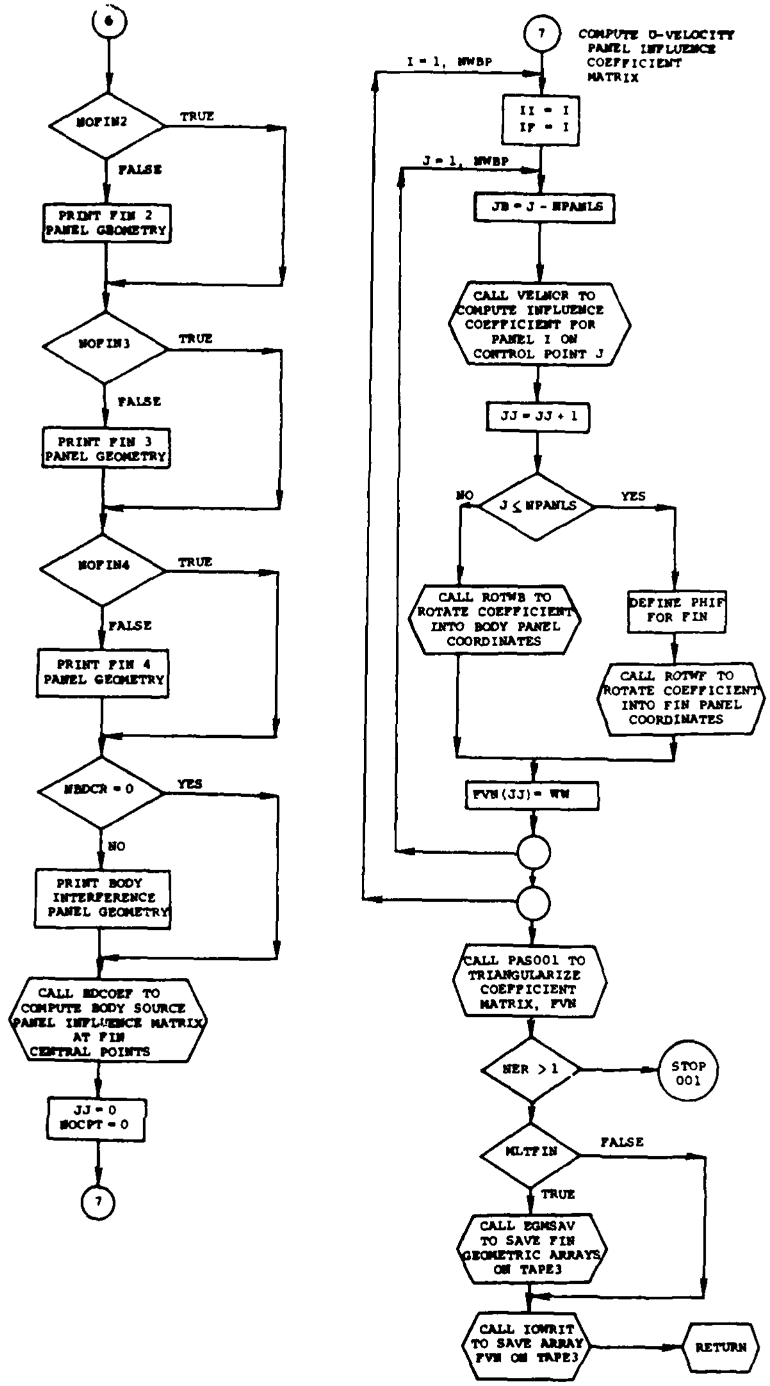
(a)

Figure C-5.- Flow chart of subroutine CRFWBD.



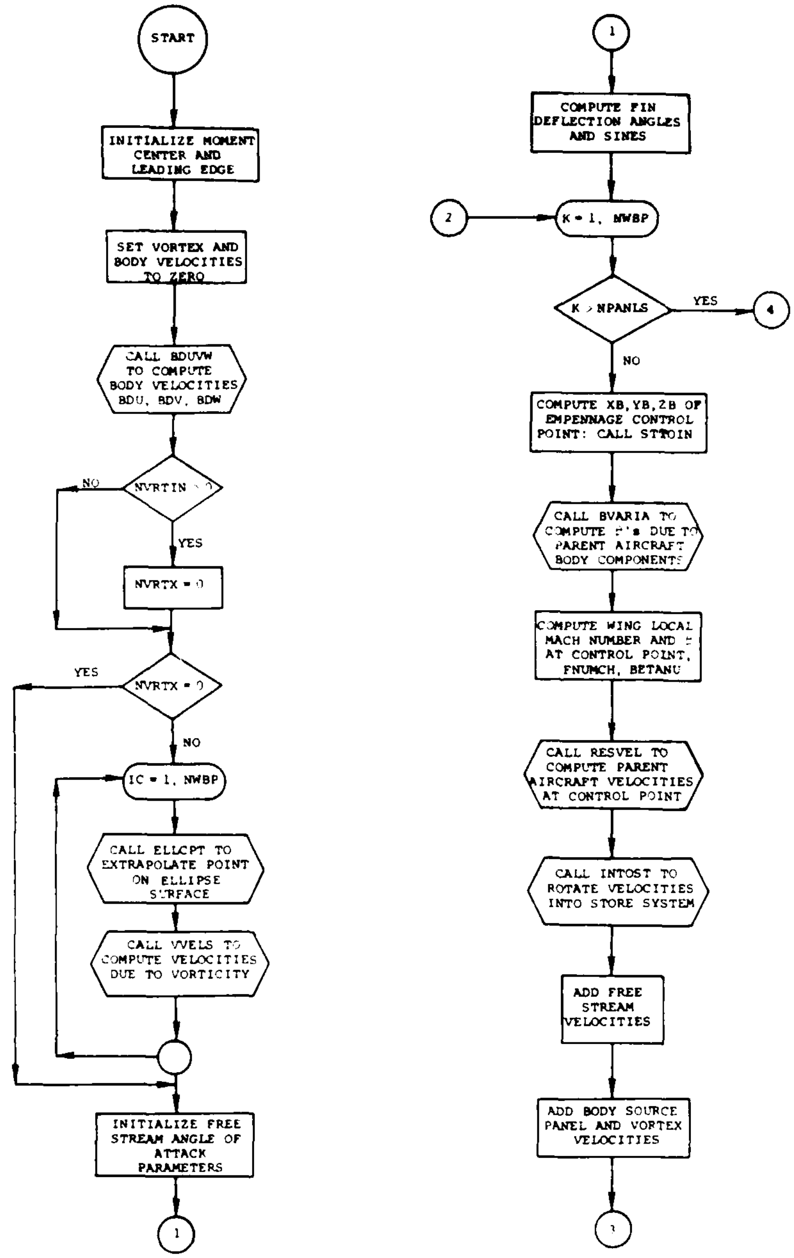
(b)

Figure C-5.- Continued.



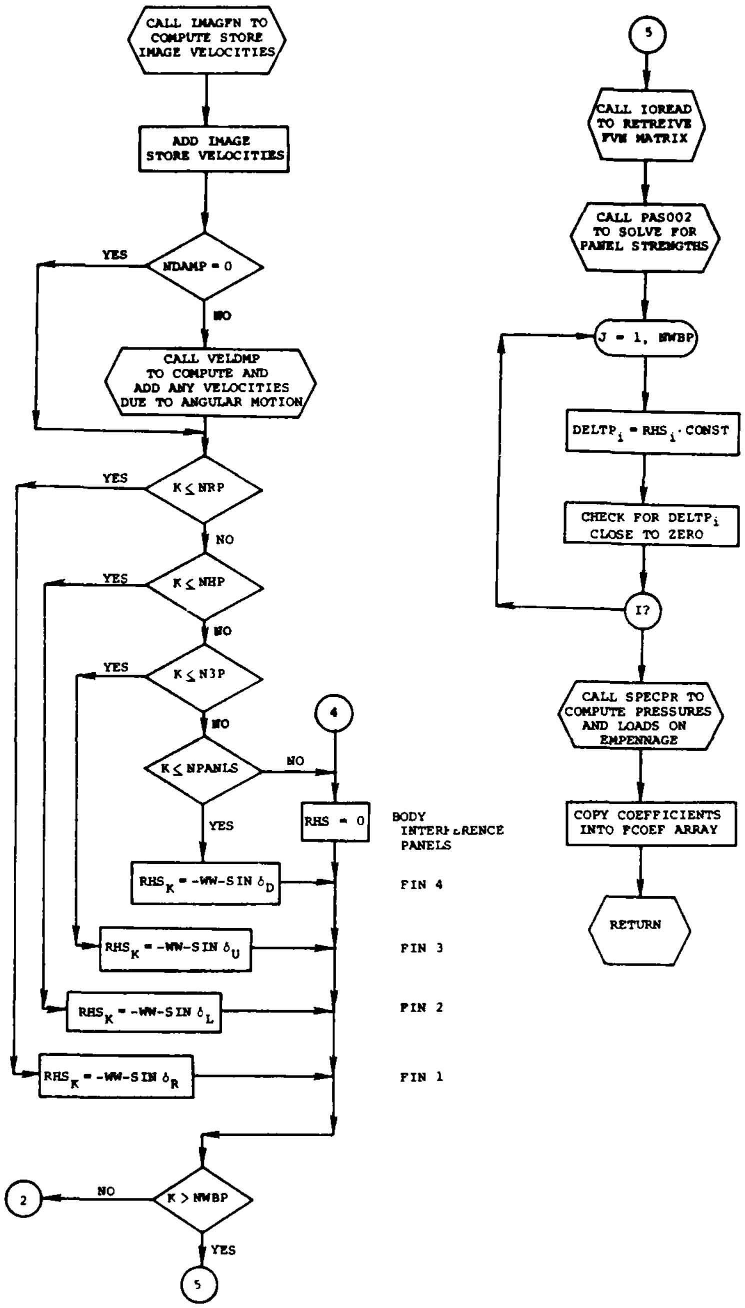
(c)

Figure C-5.- Concluded.



(a)

Figure C-6.- Flow chart of subroutine DEMON2.



(b)

Figure C-6.- Concluded.

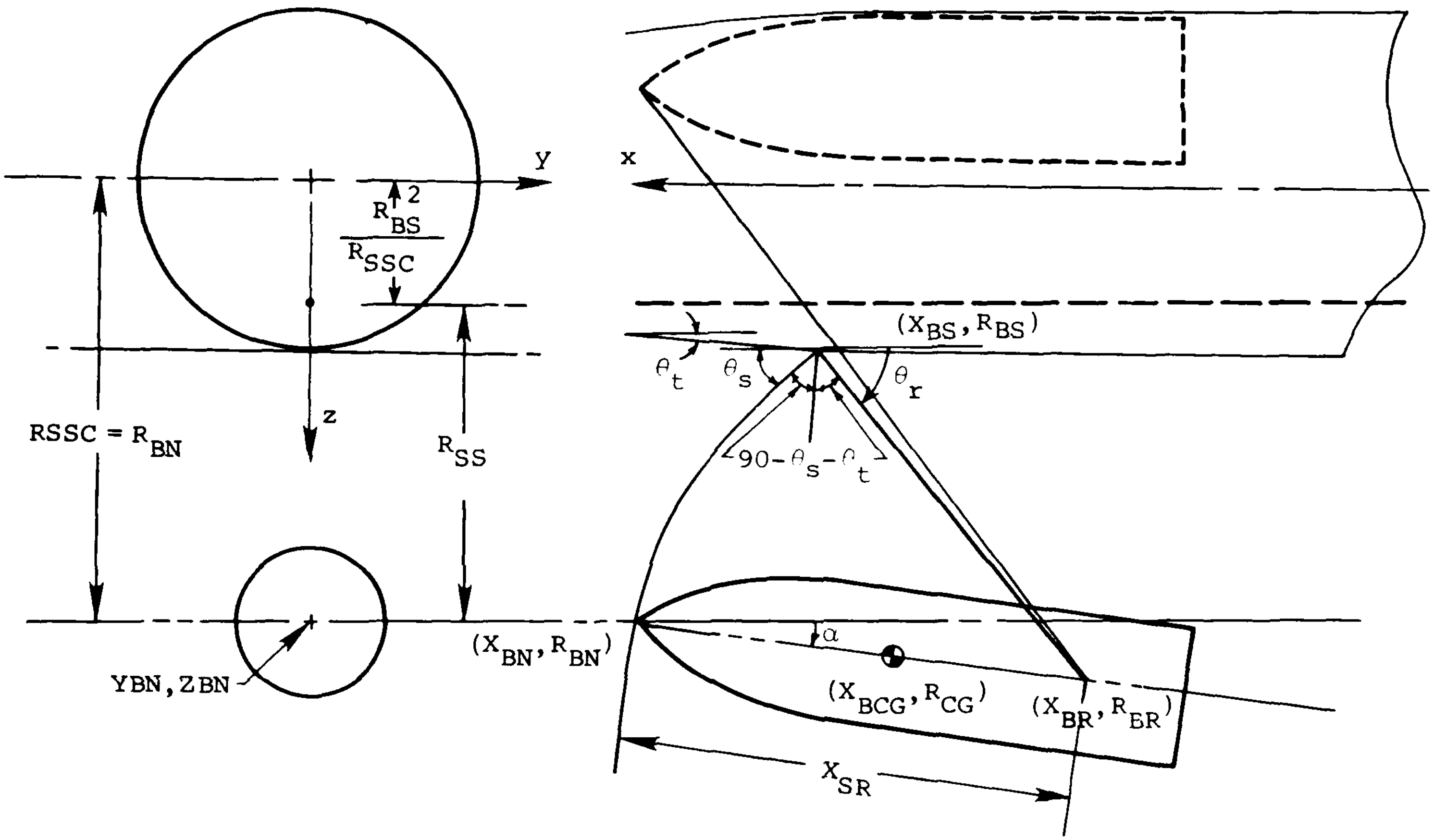
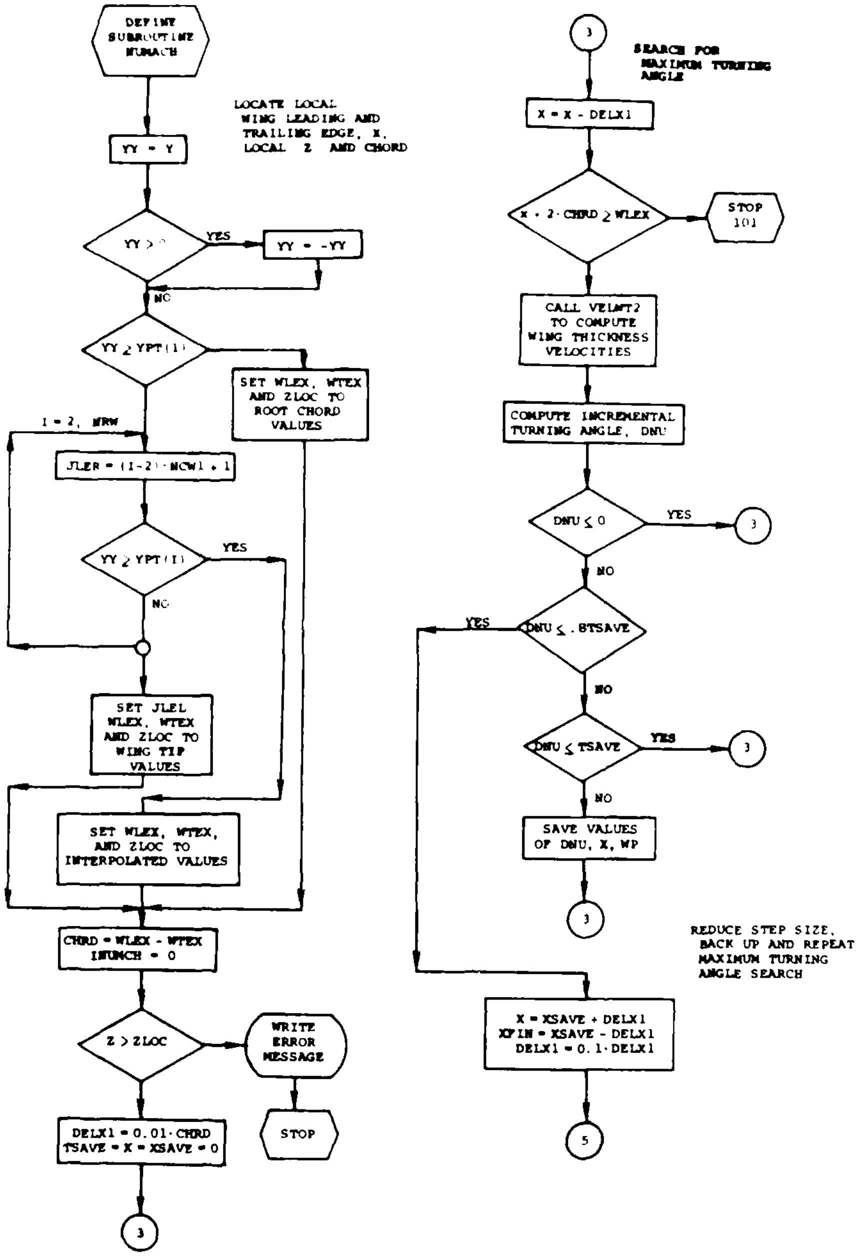
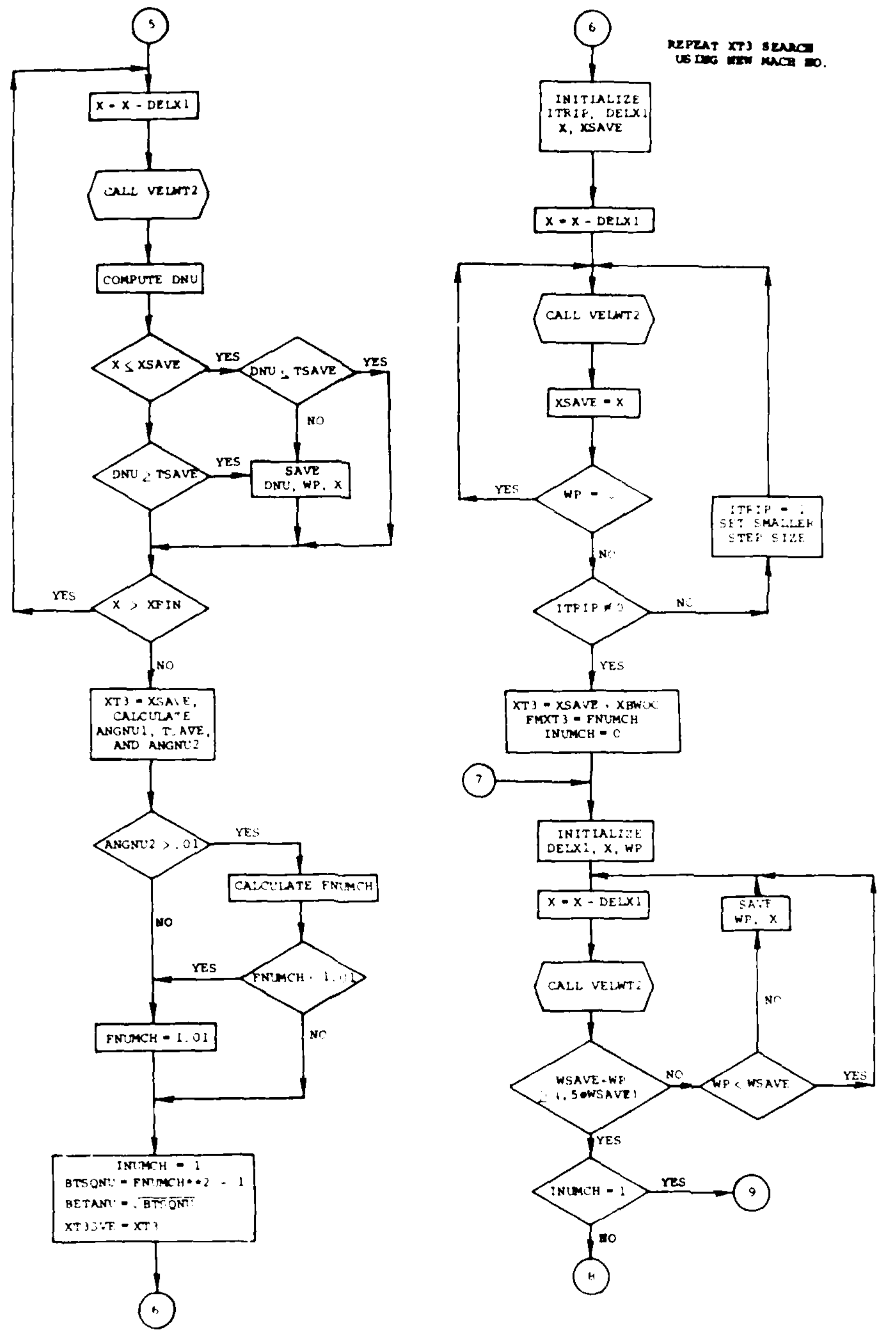


Figure C-7.- Location of image store and reflected shock from fuselage.



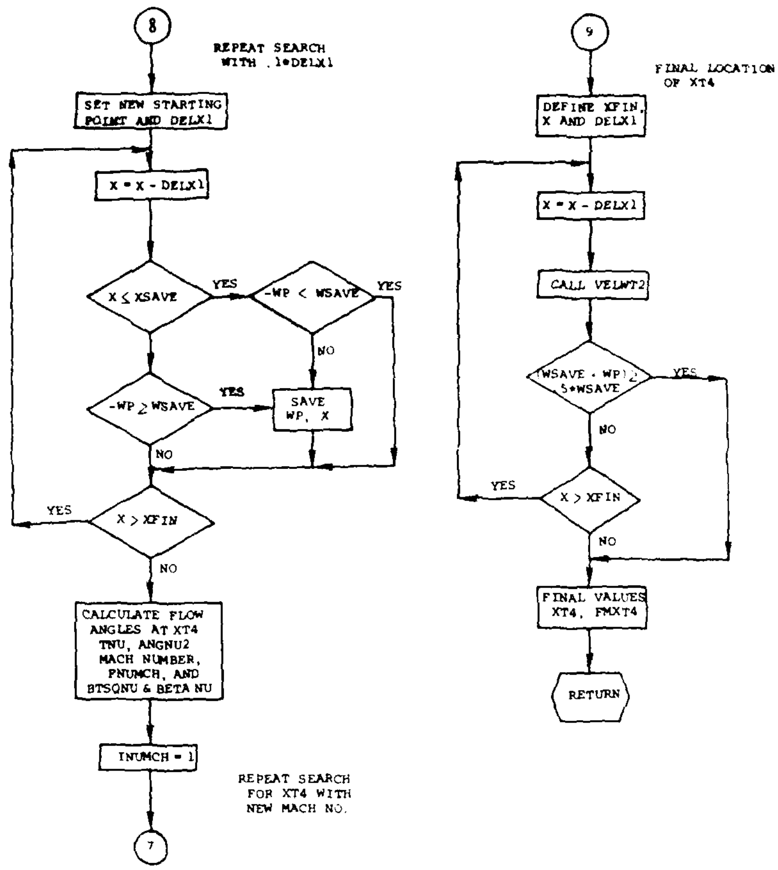
(a)

Figure C-8.- Flow chart of subroutine NUMACH.



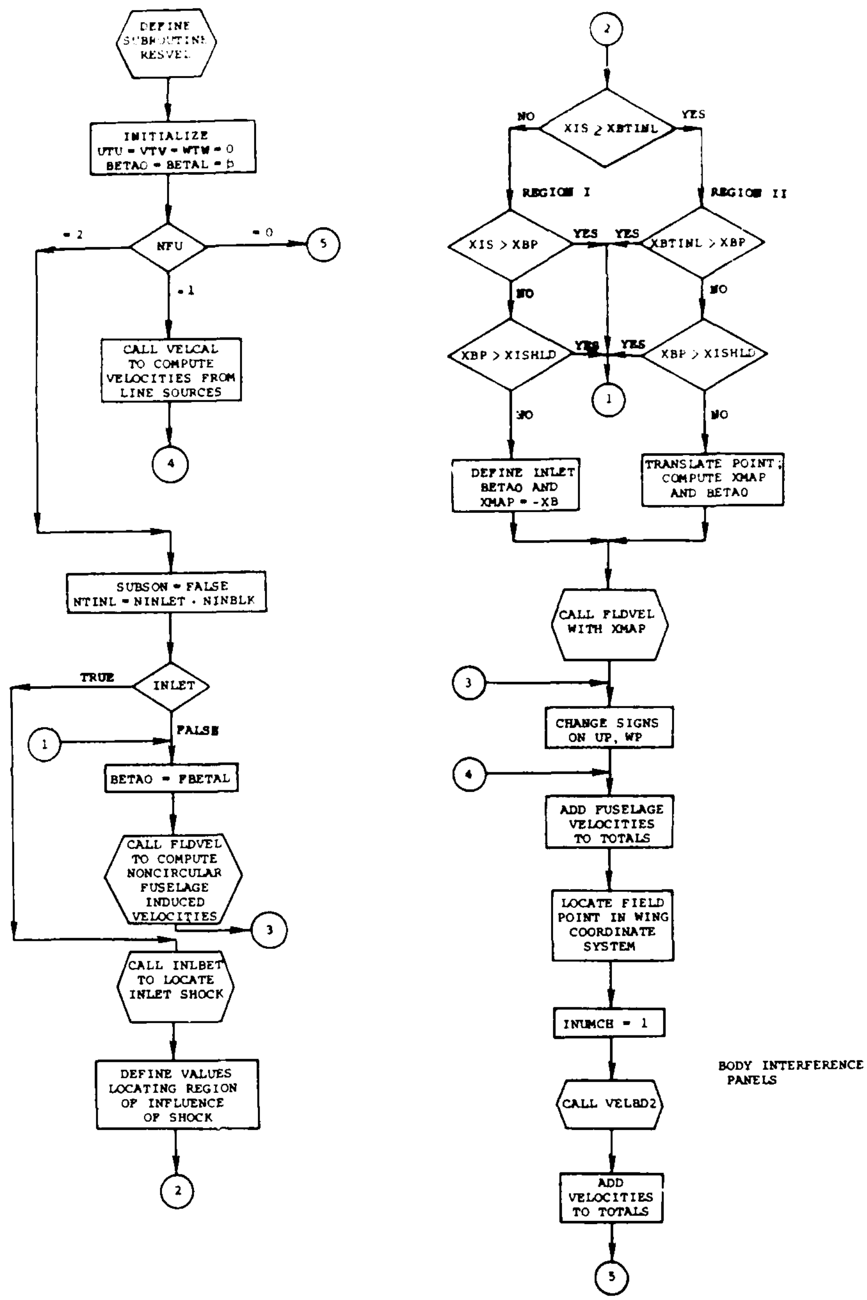
(b)

Figure C-8.- Continued.



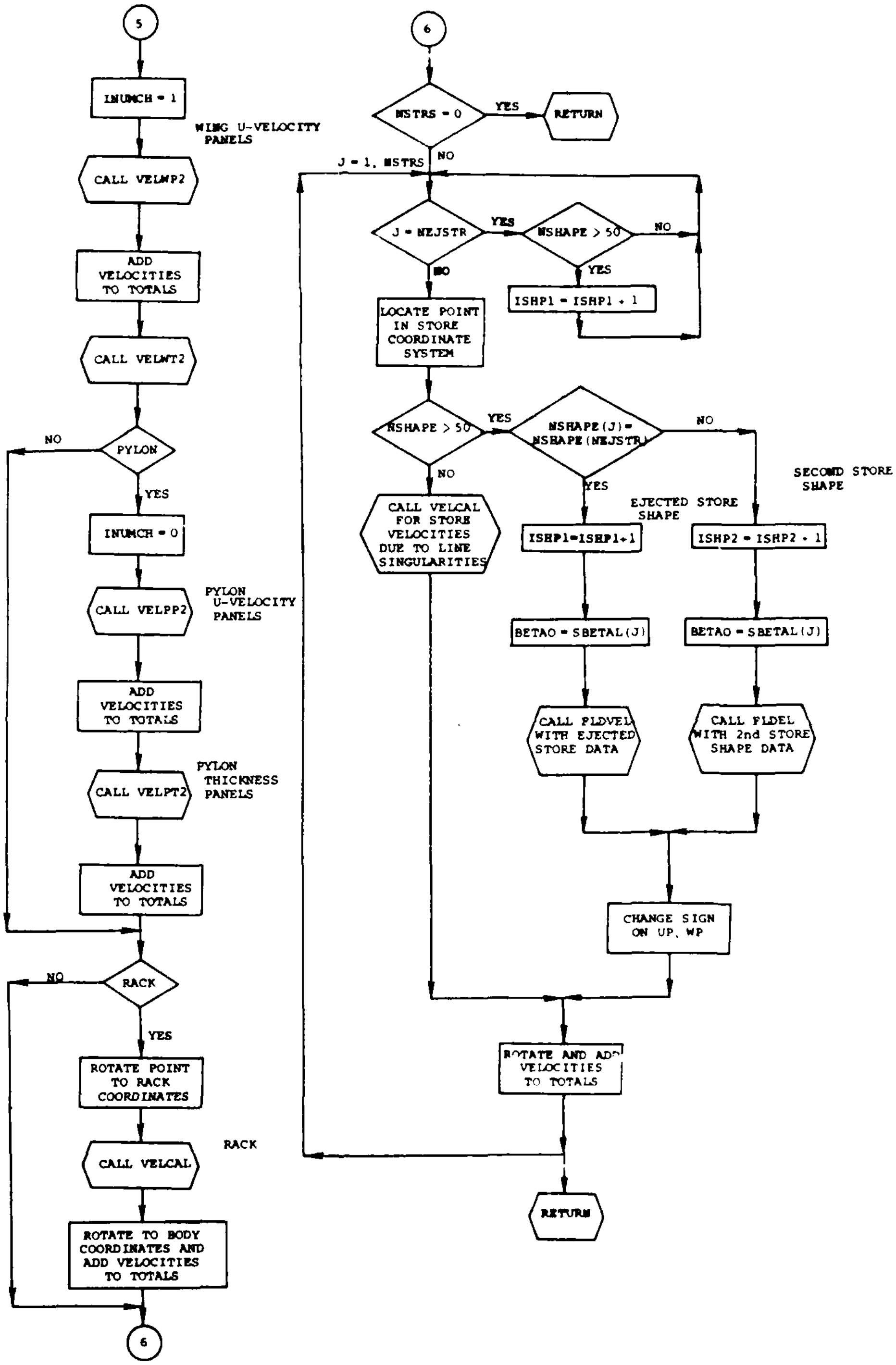
(c)

Figure C-8.- Concluded.



(a)

Figure C-9.- Flow chart of subroutine RESVEL.



(b)

Figure C-9.- Concluded.

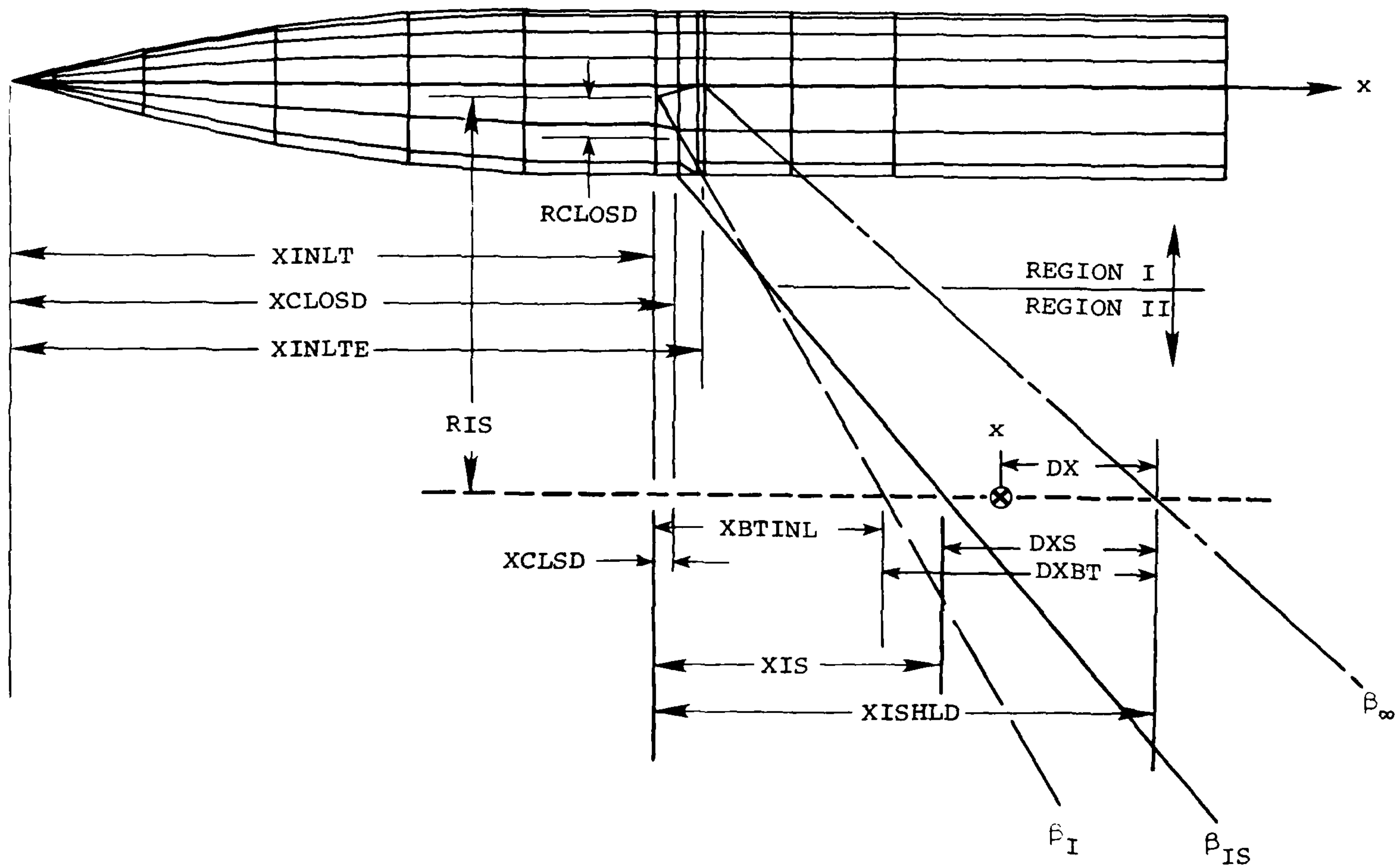
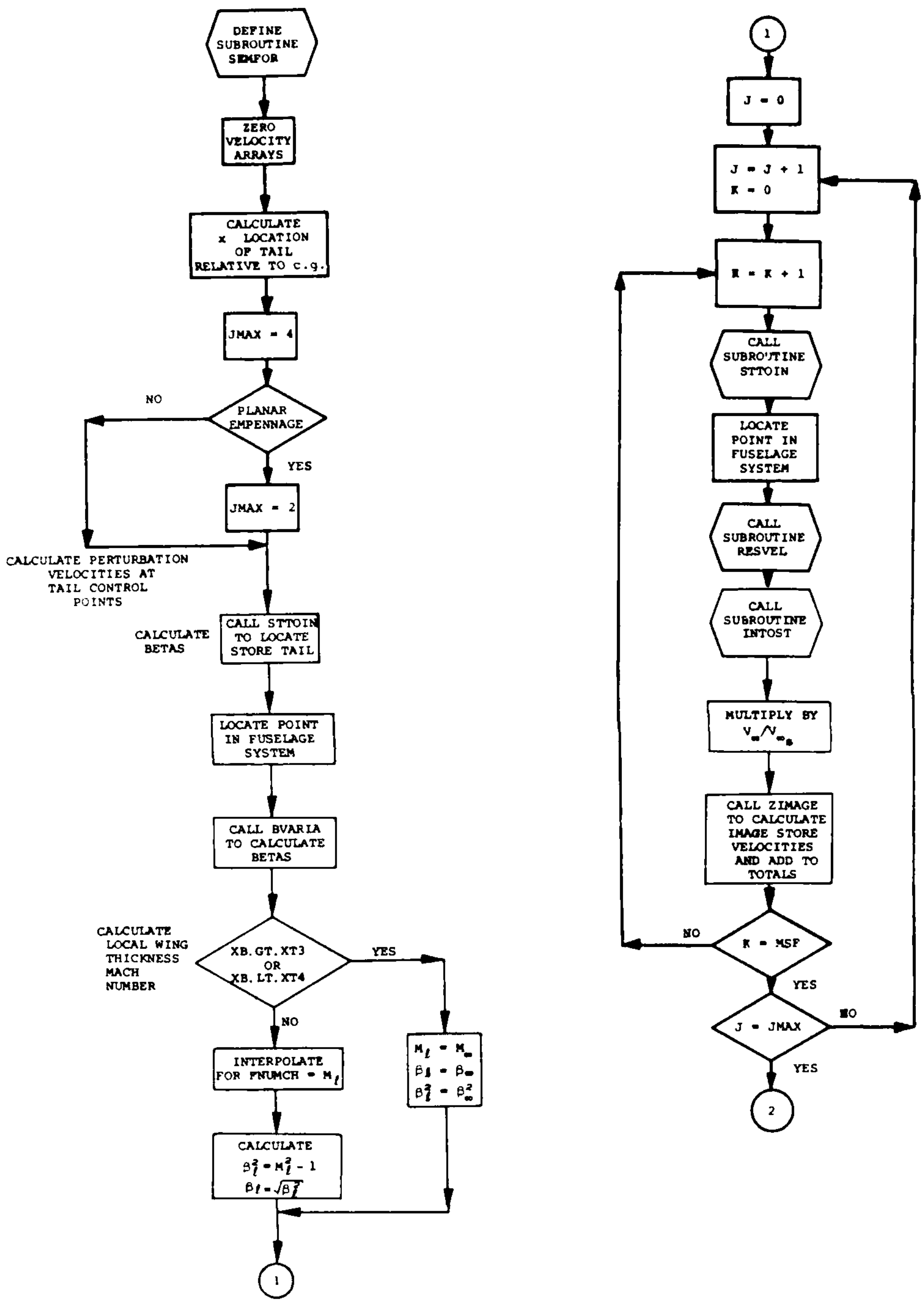
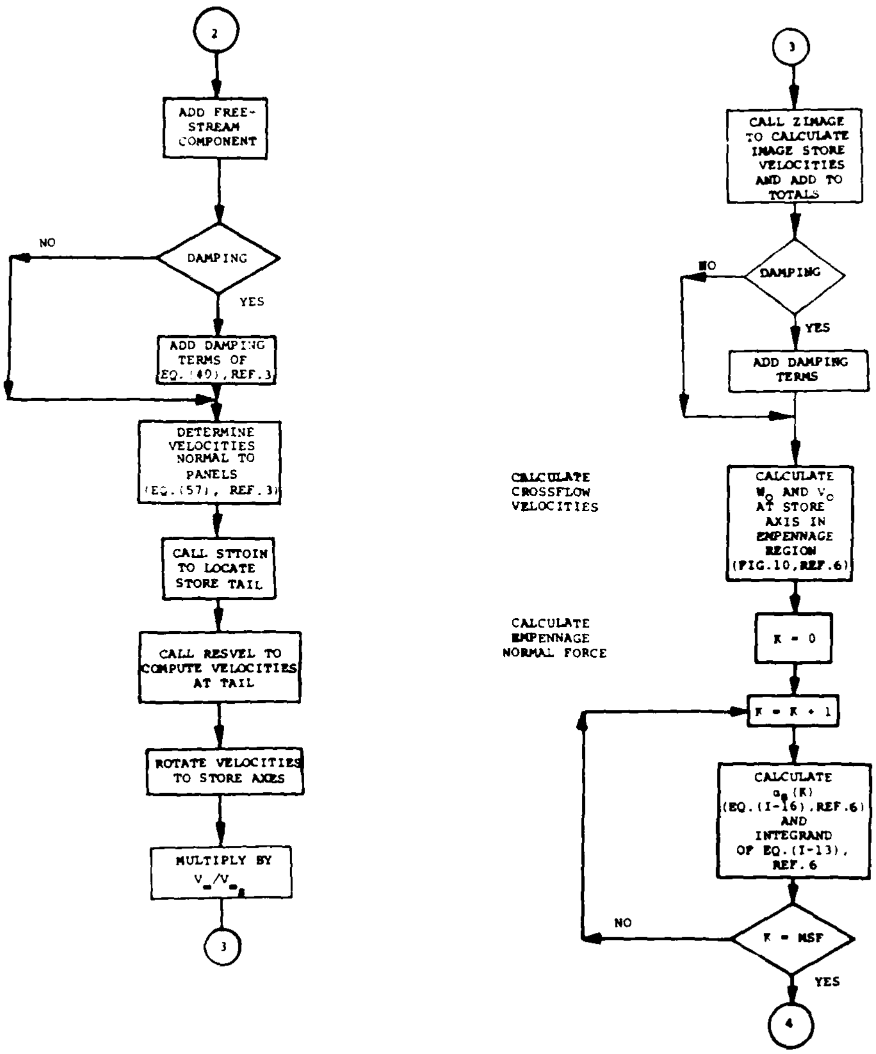


Figure C-10.- Inlet shock region of influence below fuselage.



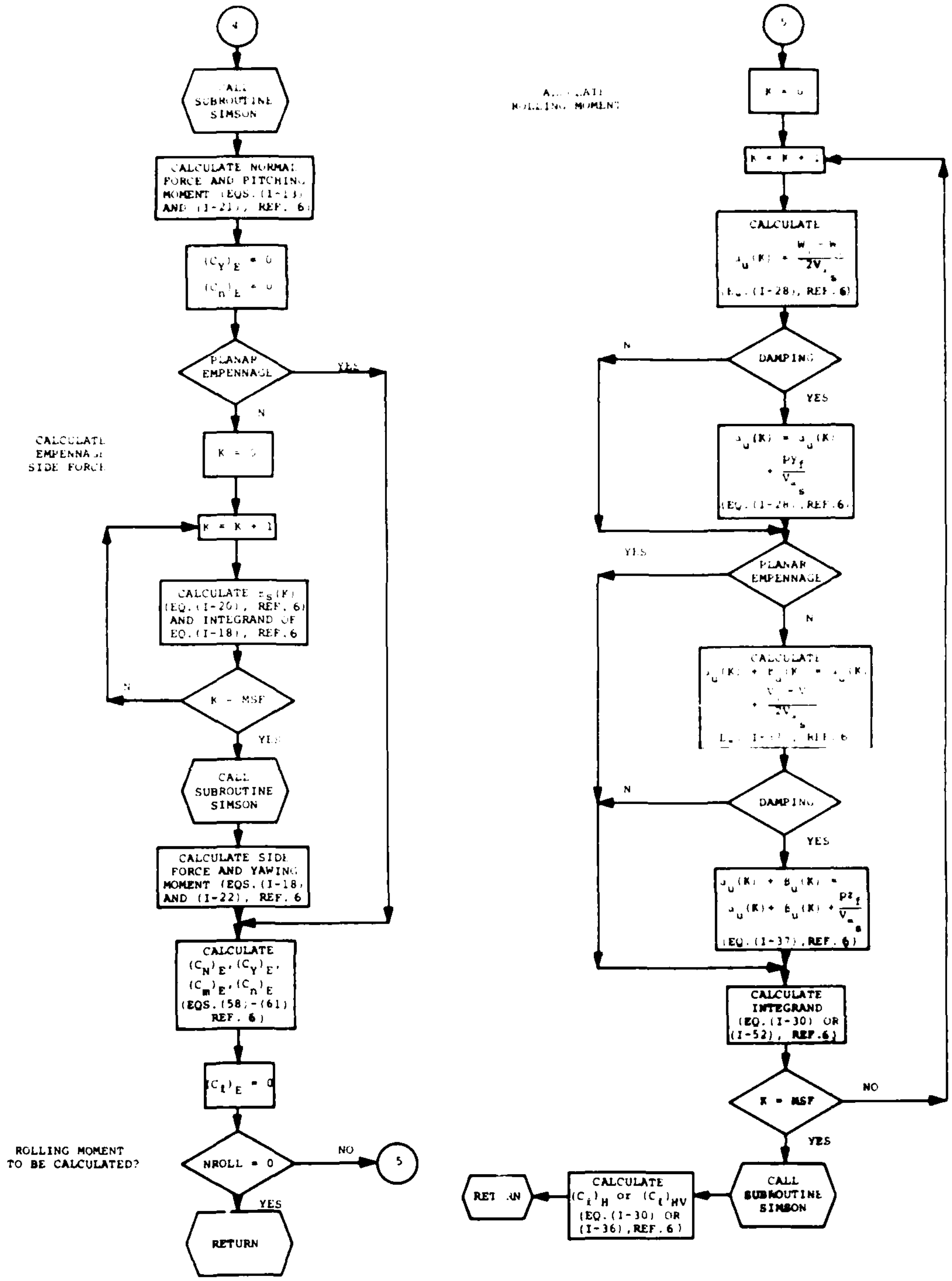
(a)

Figure C-11.- Flow chart of subroutine SEMFOR.



(b)

Figure C-11.- Continued.



(c)

Figure C-11.- Concluded.

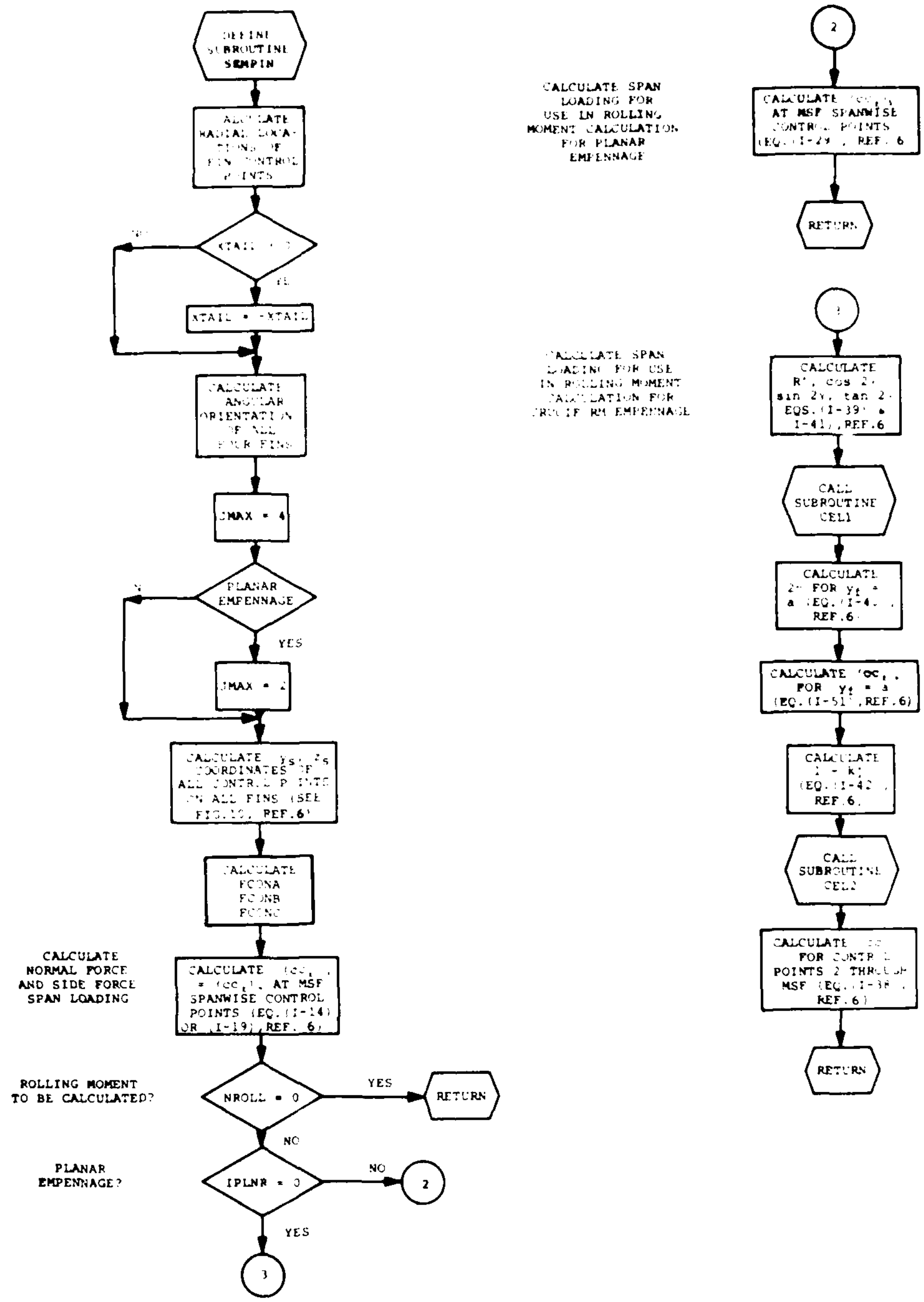
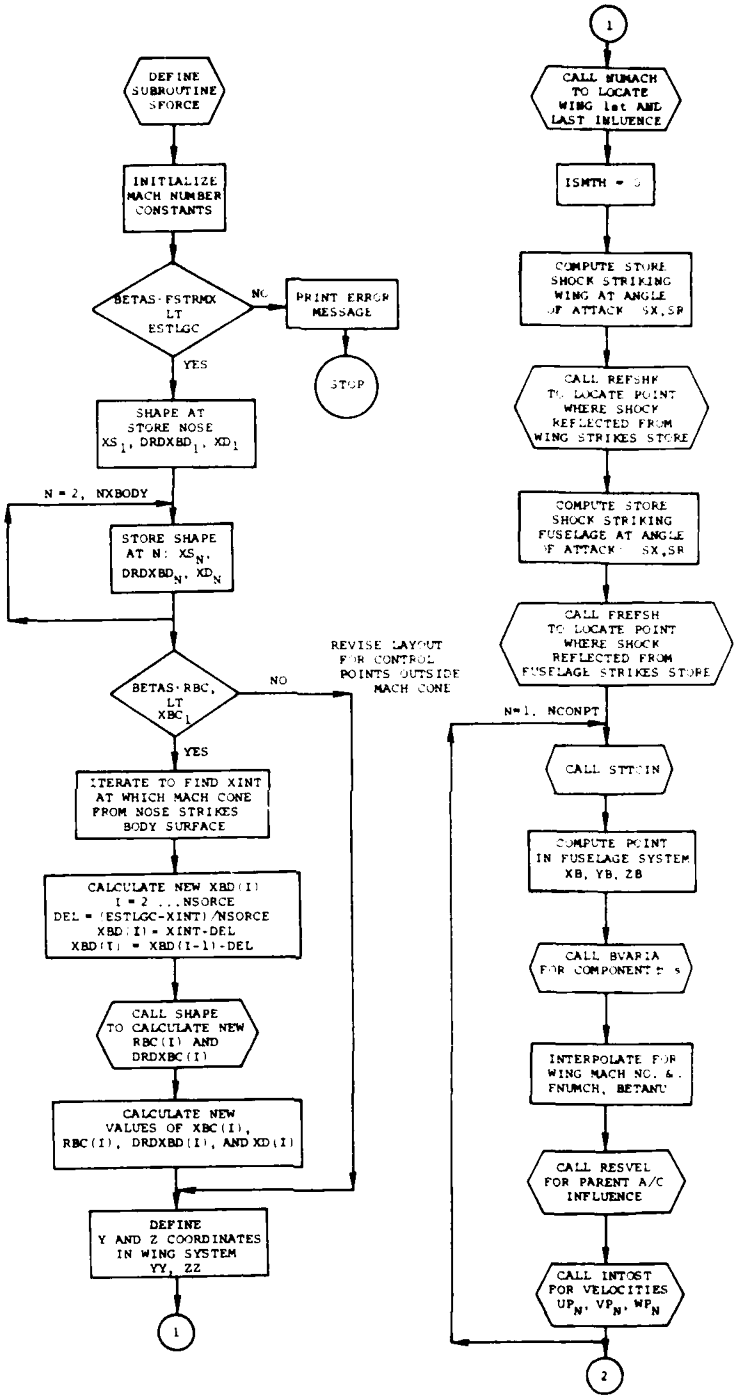
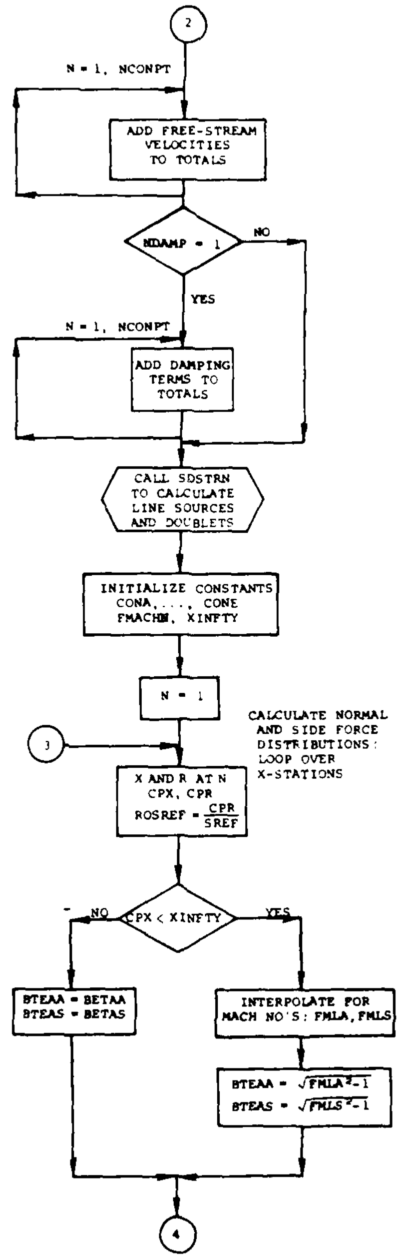


Figure C-12.- Flow chart of subroutine SEMPIN.



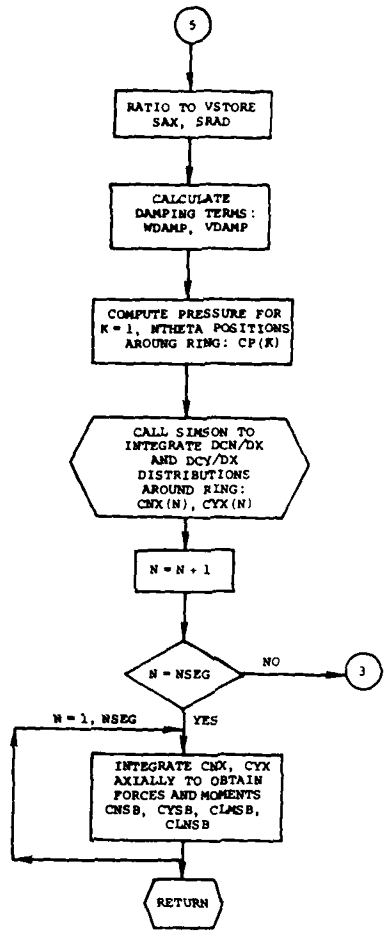
(a)

Figure C-13.- Flow chart of subroutine SFORCE.



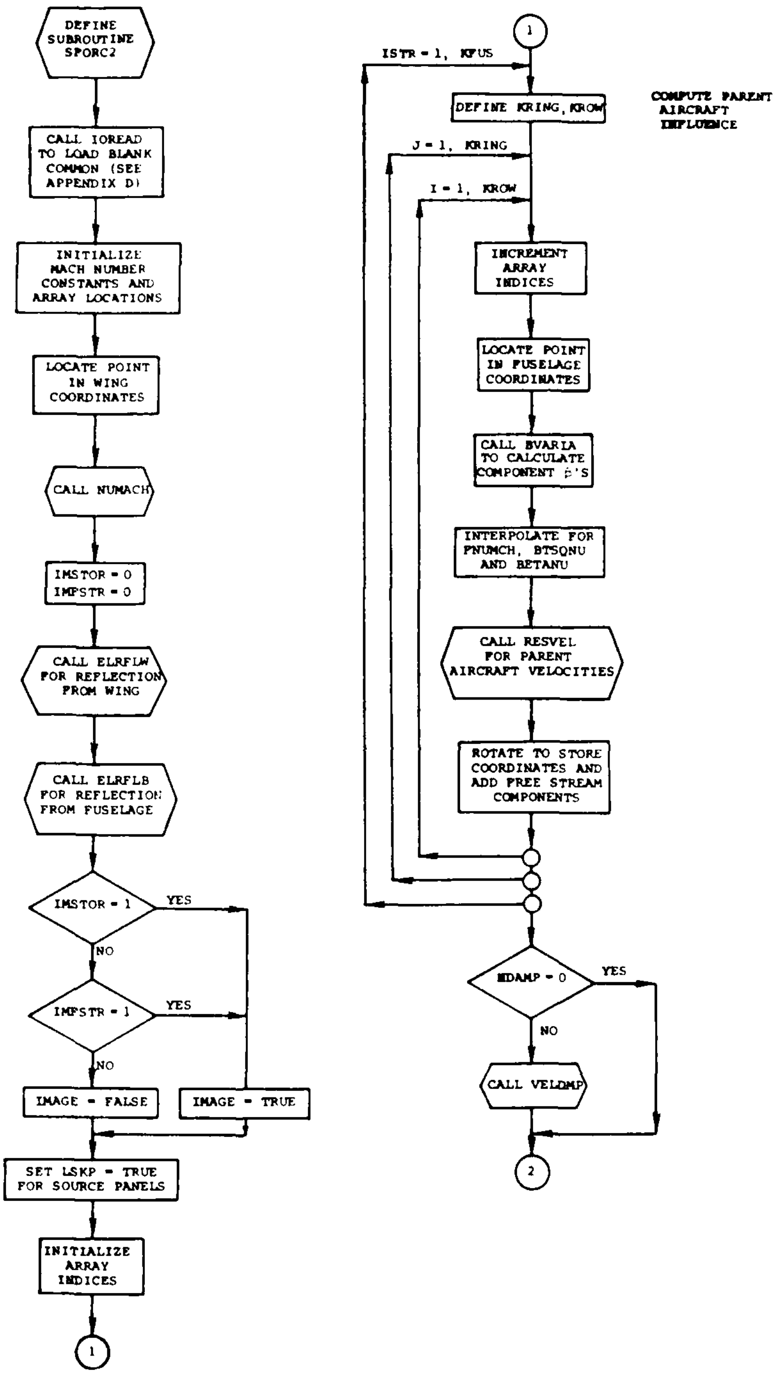
(b)

Figure C-13.- Continued.



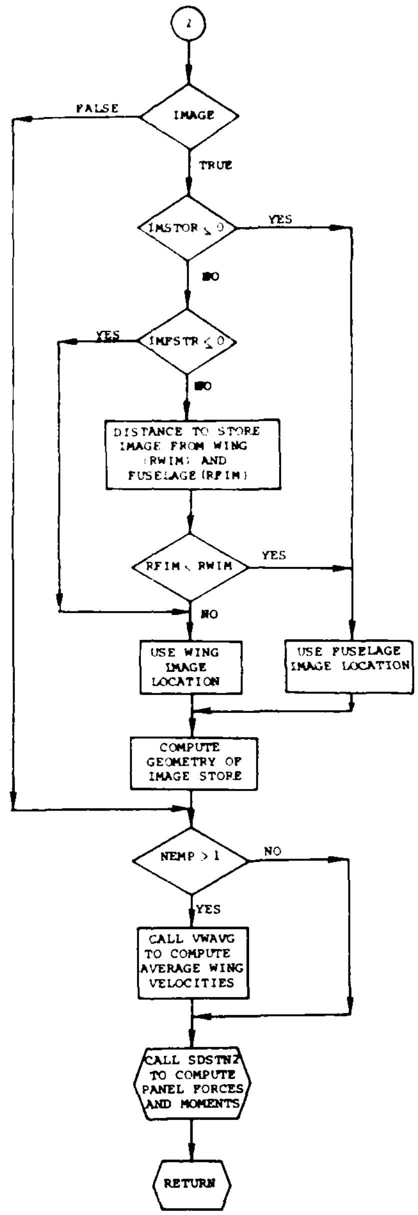
(c)

Figure C-13.- Concluded.



(a)

Figure C-14.- Flow chart of subroutine SFORC2.



(b)

Figure C-14.- Concluded.

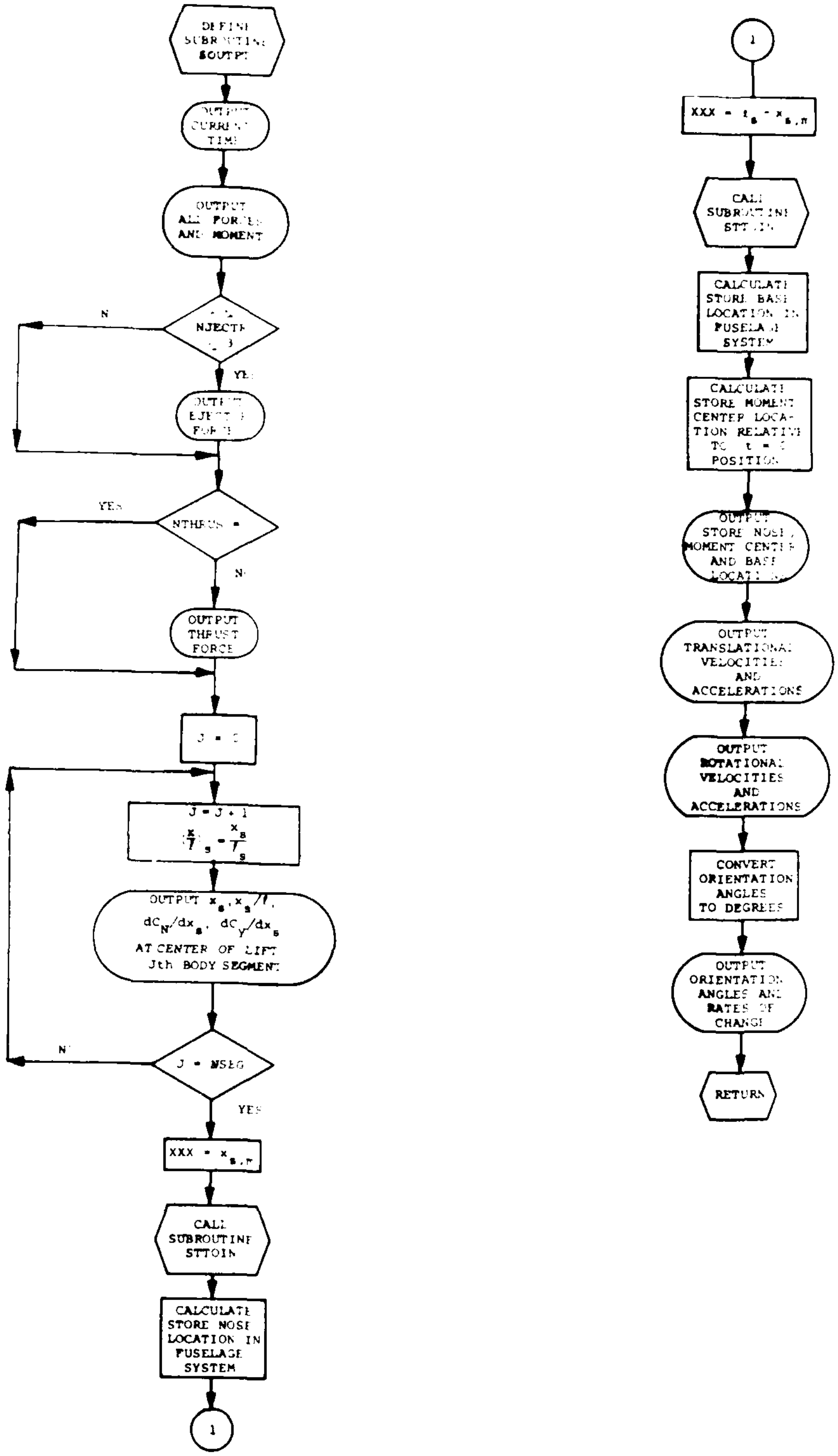


Figure C-15.- Flow chart of subroutine SOUTPT.

APPENDIX D

COMMON BLOCK DESCRIPTIONS - PROGRAM II

D-1 Introduction

The purpose of this Appendix is to provide more detailed information on the variables passed between routines through common blocks in Program II. This appendix will present the tables equating program notation to the algebraic notation and variable descriptions. The common blocks are arranged alphabetically. It is followed by a section detailing the special usage of blank common. When a description identifies an item as an input, consult Volume II of this report for further definitions.

A cross-reference chart showing the routines and the common statements contained in each is presented in Figure D-1. Across the top of the chart are the subroutine names including the main program TRJTRY. Down the side of the page are the common names. The last one is blank common.

D-2 Description of Variables in Labeled Commons

Var.	Engr. Symbol and Description

COMMON /BASESG/ FBASE,FSUMK,FSUMKD,RBASE,RSUMK,RSUMKD,SBASE(7),
SSUMK(7),SSUMKD(7)

FBASE	fuselage x-station at which base singularities originate, feet
FSUMK	strength of source originating at fuselage base
FSUMKD	strength of doublet originating at fuselage base
RBASE	rack x-station at which base singularities originate, feet

RSUMK	strength of source originating at rack base
RSUMKD	strength of doublet originating at rack base
SBASE(I)	store x-station at which base singularities originate, feet
SSUMK(I)	strength of source originating at store base
SSUMKD(I)	strength of doublet originating at store base

COMMON /BFORC/ FBOD(5,5),XBOD(5),IBOD(5),LFIN(5),NBOD

FBOD(I,J)	elliptic body force coefficients (C_Y,C_N,C_ℓ, C_m,C_n) acting on Jth body section due only to source panels
XBOD(J)	input item 16
IBOD(J)	index of panel edges in axial direction closest to value of XBOD(J) for Jth body section
LFIN(J)	logical indicator of presence of finned body section (T) or body alone section (F); input item 16
NBOD	number of sections; input item 16

COMMON /BFORCE/ CNSB,CYSB,CLMSB,CLNSB,CLLSB

CNSB	$(C_N)_{SB}$, total store body normal-force coefficient
CYSB	$(C_Y)_{SB}$, total store body side-force coefficient
CLMSB	$(C_m)_{SB}$, total store body pitching-moment coefficient
CLNSB	$(C_n)_{SB}$, total store body yawing-moment coefficient
CLLSB	$(C_\ell)_{SB}$, total store body rolling-moment coefficient
	The forces and moments in this common are those due to the store body alone in the presence of the parent aircraft and do not include any carry over induced by the presence of any

empennages. All forces and moments act in the sense displayed in Figure 17 of Volume II. All moments are about the store moment center.

COMMON /BFORCX/ CLX(50),CMX(50),CNX(50),DXC(50)

CLX(J)	C_ℓ, store body rolling-moment coefficient due to Jth ring of panels
CMX(J)	C_m, store body pitching-moment coefficient due to Jth ring of panels
CNX(J)	C_n, store body yawing-moment coefficient due to Jth ring of panels
DXC(J)	width of Jth ring of panels in axial direction

Moments in this common are those due to the store body alone in the presence of the parent aircraft and do not include any carry over induced by the presence of any empennages. All moments act in the sense displayed in Figure 17 of Volume II. All moments are about the store moment center.

COMMON /BGEOM/ XFUS(51),ZFUS(51),FUSARD(51),FUSBY(51),
FUSAZ(51),XJ(51),PHIK(33)

XFUS(I)	x_B coordinate of Ith station used to define body external geometric shape, feet; input item 18, Program I
ZFUS(I)	z_B coordinate of Ith station containing cambered offset, feet; input item 19, Program I
FUSARD(I)	cross sectional area at Ith station, ft^2; input item 22, Program I
FUSBY(I)	b_y elliptic horizontal semi-axis (y-direction), feet; input item 23, Program I
FUSAZ(I)	a_z elliptic vertical semi-axis (z-direction), feet; input item 24, Program I
XJ(J)	x_B coordinate of Jth station used to define body panel corners, feet; input item 32, Program I
PHIK(J)	ϕ_k polar angle defining meridian for Jth of panel edges, degrees; input item 31, Program I

COMMON /BINLET/ NINLET,NINVEL,NTINL,RVIVO,NINBLK,BTINLT,YCPI,
XINLT,YINLT,ZINLT,XINLTE,YINLTE,ZINLTE,JINLT(25)

NINLET	number of open inlet panels; input item 14, Program I
NINVEL	number of additional panels to be used in velocity calculations for inlet panels; input item 14, Program I
NTINL	total number of inlet panels to be used in a given calculation
RVIVO	inlet mass flow ratio; ratio of open inlet panel frontal area to total inlet panel area
NINBLK	number of blocked inlet panels; input item 14, Program I
BTINLT	β associated with inlet panels; input item 29, Program I
YCPI	y-location of inboard most edge of inlet panels
XINLT, YINLT, ZINLT	coordinates of outboard leading edge of inlet panels used to locate center of inlet shock propagation
XINLTE, YINLTE, ZINLTE	coordinates of outboard trailing edge of inlet panels used to locate lower lip of inlet and turning point of shock
JINLT(I)	fuselage panel number associated with Jth inlet panel

COMMON /BINSHK/ NIS,XISHLD,EALPI,XCLOSD,MAXSHI,NINL(8),PHINL(8),
YINL(8),XINL(80),RINL(80)

NIS	number of inlet shock traverses; input item 28, Program I
XISHLD	x-station at which β's associated with inlet return to free stream
EALPI	angle of attack correction factor for inlet shock (=EALPHA)
XCLOSD	x-location of leading edge of blocked inlet panels

MAXSHI	maximum number of points in inlet shock tables per traverse (=MAXSHK)
NINL(I)	number of points computed for Ith shock traverse
PHINL(I)	angle measured from z-axis below inlet of Ith inlet shock traverse positive counterclockwise
YINL(I)	y-station of Ith inlet shock traverse
XINL(I)	table of x-values of inlet shock
RINL(I)	table of radial values of inlet shock

```
COMMON /BLKPAN/ COST,SINT,XBTJ,YBTJ,ZBTJ,XC1,YC1,ZC1,XPTI,
     YPTI,ZPTI,COSTI,SINTI,COSD,SIND,LZERO
```

COST,SINT	$\cos(\theta_j)$ and $\sin(\theta_j)$, cosine and sine of polar angle of Jth influencing source panel
XBTJ	XPT_j coordinate of Jth influencing control point, feet
YBTJ	YPT_j coordinate of Jth influencing control point, feet
ZBTJ	ZPT_j coordinate of Jth influencing control point, feet
XC1	x_B coordinate of Jth panel reference corner, feet
YC1	y_B coordinate of Jth panel reference corner, feet
ZC1	z_B coordinate of Jth panel reference corner, feet
XPTI	XFT_i coordinate of Ith influenced field point, feet
YPTI	YFT_i coordinate of Ith influenced field point, feet
ZPTI	ZFT_i coordinate of Ith influenced field point, feet
COSTI,SINTI	$\cos(\theta_i)$ and $\sin(\theta_i)$, cosine and sine of polar angle at Ith influenced panel or field point

COSD,SIND	$\cos(\delta_i)$ and $\sin(\delta_i)$, cosine and sine of panel incidence angle at Ith influenced panel or field point
LZERO	PANVEL angle calculation option: F=yes, T=no

COMMON /BLK1/ BY,AZ,R,S,PI,CI

BY	local elliptic store horizontal semi-axis
AZ	local elliptic store vertical semi-axis
R	average of elliptic semi-axes; = $\frac{1}{2}$(BY+AZ)
S	local fin span set to horizontal semi-axis (=BY)
PI	π
CI	complex one: (=COMPLEX(0,1))

COMMON /BODCOM/ AMACH,TAND,CX,XCOR(4),YCOR(4),ZCOR(4),XI,YI,ZI,
XJ,ZJ,BETA0,BETAL,SUBSON,SUPERS

AMACH	Mach number used in source panel influence calculation
TAND	$\tan\delta_j$, tangent of incidence angle of Jth influencing panel
CX	panel chord length, feet
XCOR(K)	x of Kth corner in local panel system, feet
YCOR(K)	y of Kth corner in local panel system, feet
ZCOR(K)	z of Kth corner in local panel system, feet
XI	x of Ith field point in local panel system, feet
YI	y of Ith field point in local panel system, feet
ZI	z of Ith field point in local panel system, feet
XJ	x of Jth panel control point in panel system, feet
ZJ	z of Jth panel control point in panel system, feet

BETA0	$\sqrt{M^2-1}$, Mach number constant corrected for body nose or inlet shock location
BETAL	$\sqrt{M_\ell^2-1}$, local Mach number constant used in influence calculation
SUBSON	subsonic logical indicator; SUBSON = AMACH.LT.1
SUPERS	supersonic logical indicator; SUPERS = AMACH.GT.1

COMMON /BOPTNS/ J0,J2,J6,NFUS,NRADX(5),NFORX(5),J2TEST,IPRES,ISOLV, INLET,IPLOT(4),IPRT(5),IUVW,XSTART,XWLE,REFA,REFD,REFL,REFX,REFZ, CCTEST,ITMAX,BODL,IZ1(12)

J0	reference area indicator; input item 16, Program I
J2	body type indicator; input item 16, Program I
J6	body camber indicator; input item 16, Program I
NFUS	number of body segments; input item 16, Program I
NRADX(I)	number of points used to define section of Ith body segment; input item 16, Program I
NFORX(I)	number of axial station on Ith body segment; input item 16, Program I
J2TEST	parameter to specify body camber and cross section definition
IPRES	not used
ISOLV	not used
INLET	logical inlet indicator: True=inlet panels present, False=no inlet panel present
IPLOT(I)	not used
IPRT(I)	optional print control parameter, input item 14, Program I
IUVW	component velocity calculation option; input item 14, Program I
XSTART	x-station at which pressure integration is started, feet
XWLE	x-station at which pressure integration is ended, feet

REFA	body reference area, ft^2; input item 17, Program I
REFD	body reference length used for moment normalization, feet; input item 30, Program I
REFL	body length, feet; input item 30, Program I
REFX,REFZ	x,z coordinates of moment reference point, feet; input item 30, Program I
CCTEST	solution convergence control criteria (=0.0001)
ITMAX	solution maximum number of iterations (=20)
BODL	body length over which panels are laid out, feet
IZ1(I)	dummy array, not used

COMMON /BPHII/ ZIP,PHII,CPHI,SPHI,C2PHII,S2PHII

ZIP	distance between elliptic store nose and image store nose
PHII	angle between fuselage plane of symmetry and line between noses of real and image stores
CPHI,SPHI	cos ϕ_I and sin ϕ_I, ϕ_I = PHII - ϕ, angle between elliptic store vertical axis and line between noses of real and image stores
C2PHII,S2PHII	cos $2\phi_I$ and sin $2\phi_I$

COMMON /BSHOCK/ NSHK(10),PHIS(10),THETN(10),MAXSHK,NSHOCK,DBETA,
EALPHA,CNU0,CNU2,XSHLDR,SHK(3),XSHK(100),RSHK(100)

NSHK(I)	number of points used to represent Ith modified shock shape
PHIS(I)	polar angle at which Ith modified shock is computed, degrees; input item 33, Program I
THETN(I)	nose limited shock angle of initial shock shape at Ith polar angle, degrees
MAXSHK	maximum number of points in Ith shock shape; input item 14, Program I
NSHOCK	number of modified shock shape computed; input item 14, Program I
DBETA	not used

EALPHA	angle of attack correction to shock shape; input item 15, Program I
CNU0,CNU2	not used
XSHLDR	x_B location of body nose shoulder, feet; input item 15, Program I
SHK(I)	dummy array, not used
XSHK(I),RSHK(I)	arrays containing x_B and r_B locations of NSHOCK sets of NSHK(I) points representing the modified nose shock shape; feet

COMMON /BSWINT/ IP,XPTIP,YPTIP,ZPTIP,DELTIP,DRDXS,YBS,ZBS,BTNOSN

IP	fuselage source panel number of first intersection of store nose shock with noncircular fuselage body
XPTIP, YPTIP, ZPTIP	x,y,z coordinates in source panel system of panel IP
DELTIP	δ_{IP}, incidence angle of panel IP
DRDXS	dr/dx_S, slope of store nose shock at intersection with fuselage in plane between fuselage centerline and store nose
YBS,ZBS	y_B, z_B coordinates of intersection point of store nose shock with fuselage
BTNOSN	β, Mach number parameter computed from nose of image store to intersection of reflected store shock from fuselage with store body centerline

COMMON /BVEL/ BDU(250),BDV(250),BDW(250)

BDU(I), BDV(I), BDW(I)	u,v,w components of velocity induced by elliptic store body source panels at empennage u-velocity control points. All velocities act in the direction of u_S,v_S,w_S shown in Figure 17 of Volume II.

COMMON /BVELFS/ VX,VY,VZ,VRATS,RVX,RVY,RVZ

VX	$U_{\infty s,x_s}$ Equation (93), Reference 2
VY	$-V_{\infty s,y_s}$ Equation (93), Reference 2

VZ	$W_{\infty s,z_s}$ Equation (93), Reference 2
VRATS	$V_\infty/V_{\infty s}$, ratio of parent aircraft to store velocity
RVX	VX/VRATS
RVY	VY/VRATS
RVZ	VZ/VRATS

`COMMON /BVELO/ UVEQO`

UVEQO	logical variable used to set the u,v perturbation velocities induced by one panel on another panel in the same plane equal to zero

```
COMMON /CFORCE/ CNX(80),CYX(80),XCP(81),DRDXCP(81),RCP(81),
     UT(81),VT(81),WT(81)
```

CNX(I)	total dC_N/dx_s acting at the midpoint of the Ith circular body segment or acting at the center of pressure in the normal direction on the Ith ring of panels on an elliptic store
CYX(I)	total dC_Y/dx_s acting at the midpoint of the Ith circular body segment or acting at the center of pressure in the lateral direction on the Ith ring of panels on the elliptic store
XCP(I)	for circular stores it is the midpoint of the Ith segment, for elliptic stores it is the center of pressure for the Ith store ring of panels
DRDXCP(I)	dr/dx_s of circular store radius distribution at midpoint of Ith body segment
RCP(I)	radius of circular store at midpoint of Ith body segment
UT(I), VT(I), WT(I)	U_s^*,V_s^*,W_s^*, parent aircraft velocities at Ith circular body segment

`COMMON /COM1/ A2,B2,R2`

A2	a^2, square of vertical semi-axis of ellipse
B2	b^2, square of horizontal semi-axis of ellipse

R2 — r^2, square of radius of circle in the transformed plane; Equation (I110), Reference 5

COMMON /COM2/ SIG2,H2

SIG2 — $\sigma^2 = \frac{1}{4}(a+b)^2$; Equation (I107), Reference 5

H2 — 4·R2, R2 given above; Equation (I110), Reference 5

COMMON /COM3/ ZR,ZI

ZR — real part of z, $R(z) = x$

ZI — imaginary part of z, $I(z) = y$, where x and y are the coordinates in the crossflow plane as defined in Appendix I of Reference 5

COMMON /COM4/ G2,G1

G2 — $=G1^2 - H2$, see square root term in Equation (I113), Reference 5

G1 — $= \frac{w+\sigma^2}{w}$

COMMON /COM5/ DWDZ

DWDZ — Equation (I127), Reference 5

COMMON /COM6/ W2,W

W2 — $1/W^2$

W — Equation (I112), Reference 5

COMMON /COM9/ IGROW

IGROW — index indicating variation of elliptic semi-axes, a and b, with axial location

COMMON /CONFIG/ NFU,NPY,NSTRS,NRACK

NFU — fuselage indicator; input item 4, Program I

NPY — pylon indicator; input item 4, Program I

NSTRS	number of stores indicator; input item 4, Program I
NRACK	rack indicator; input item 4, Program I

COMMON /CONSTS/ PI,DTOR

PI	π
DTOR	$\pi/180$

COMMON /COUTPT/ VAR(12),DVAR(12),TIME,DC(3,3),CDC,XMON,ESTRMX,
DX,DELX,VSTORE,ESTLGC

VAR(N), N=1,2,...12	$\dot{\xi},\dot{\eta},\dot{\zeta},p,q,r,\xi,\eta,\zeta,\Psi,\Theta,\Phi$, respectively
DVAR(N) N=1,2,...12	$\ddot{\xi},\ddot{\eta},\ddot{\zeta},\dot{p},\dot{q},\dot{r},\dot{\xi},\dot{\eta},\dot{\zeta},\dot{\Psi},\dot{\Theta},\dot{\Phi}$, respectively
TIME	t, current value of time
DC(I,J)	[A], Equation (86), Reference 2
CDC	not used
XMOM	$x_{s,m}$; input item 10
ESTRMX	maximum radius, a_{max}, of separated store
DX	DELX/2
DELX	length of body segment used in force calculation
VSTORE	store free-stream velocity, V_{∞_s}, Equation (97), Reference 2
ESTLGC	length, ℓ_s, of separated store

COMMON /DIMENS/ NX,NR,KX,KR,NXNR,KXKR,NXKR,MAXNX,MAXKR,NATOT,
NBODY,KFUS,KRADX(5),KFORX(5),IXC(5),IYC(5),IZC(5),
IXZSYM,NADIM,NXDIM,NG,IXPT,IYPT,IZPT,ITH,IDEL,NTAP7,
IAR,IAN,IUB,IGB(7),IVB,IU,IV,IW,IVA,IWA,ICP,IPHI,IYB,
NAG,NAP,NAV,NAS,NASHK,NAFLD,IA0,ID0,ISK0,IYIM,IZIM,
ISVN,ISKP,NRING,IROW(50)

NX	sum of axial geometry stations (= $\sum^{NFUS}$ NFORX(I))
NR	not used
KX	sum of axial panel geometry stations (= $\sum^{NFUS}$ KFORX(I))

KR	not used
NXNR	sum $\left(= \sum^{NFUS} NFORX(I)*NRADX(I)\right)$
KXKR	sum $\left(= \sum^{KFUS} KFORX(I)*KRADX(I)\right)$
NXKR	sum $\left(= \sum^{NFUS} NFORX(I)*KRADX(I)\right)$
MAXNX	maximum of (NFORX(I),I=1,NFUS)
MAXKR	maximum of (KFORX(I),I=1,NFUS)
NATOT	last location accessed in blank common
NBODY	number of body panels
KFUS	number of body segments paneled (=NFUS)
KRADX(I)	number of meridian lines used to define panel edges on Ith body segment; input item 27, Program I
KFORX(I)	number of axial stations used to define leading and trailing edges of panels on Ith body segment; input item 27, Program I
IXC(I)	location in blank common of start of XC array for Ith body segment
IYC(I)	location in blank common of start of YC array for Ith body segment
IZC(I)	location in blank common of start of ZC array for Ith body segment
IXZSYM	XZ-plane symmetry option; input item 14, Program I
NADIM	dimensioned length of A array in blank common
NXDIM	maximum allowable number of axial stations (=51)
NG	last location in blank common containing source panel geometry arrays
IXPT	location in blank common of start of control points, XPT
IYPT	location in blank common of start of control points, YPT
IZPT	location in blank common of start of control points, ZPT

ITH	location in blank common of start of array THET
IDEL	location in blank common of start of array DELTA
NTAP7	number of variables last written on TAPE7
IAR	location in blank common of start of panel areas, AREA
IAN	location in blank common of start of temporary array AN containing influence coefficients
IUB	location in blank common of start of temporary array UB containing U,V,W influence coefficients
IGB(IALP)	location in blank common of start of IALPth array containing the source strength solution, GB
IVB	location in blank common of start of temporary array, VB, containing normal velocity boundary conditions
IU	location in blank common of start of U velocities
IV	location in blank common of start of V velocities
IW	location in blank common of start of W velocities
IVA,IWA	locations in blank common of start of additional temporary velocity arrays
ICP	location in blank common of start of pressure coefficient, CP, array
IPHI	location in blank common of temporary PHI array
IYB	location in blank common of start of coordinates YB and ZB of temporary cross section geometry in NEWRAD
NAG	maximum locations in blank common required in GEOM
NAP	not used
NAV	maximum locations in blank common required in VELCMP
NAS	maximum locations in blank common required in SOLVE

NASHK	maximum locations in blank common required in BSHOCK
NAFLD	maximum locations in blank common required in FLDVEL
IA0	offset location in blank common of all above arrays when multiple configurations are simultaneously in core
ID0	offset location in blank common of ID array containing /DIMENS/ information for additional source panel configurations in core
ISK0	offset location in blank common of array containing information in common /BSHOCK/ for additional source panel configurations in core
IYIM,IZIM	locations in blank common of start of image store Y and Z control point coordinates
ISVN,ISKP	locations in blank common of start of temporary arrays, SVN and LSKP
NRING	number of rings of panels on body
IROW(I)	number of panels around Ith ring of panels

COMMON /EFORC1/ CYEM1,CNEM1,CLLEM1,CLMEM1,CLNEM1

CYEM1	$(C_Y)_{EM}$, store side-force coefficient due to 1st empennage
CNEM1	$(C_N)_{EM}$, store normal-force coefficient due to 1st empennage
CLLEM1	$(C_\ell)_{EM}$, store rolling-moment coefficient due to 1st empennage
CLMEM1	$(C_m)_{EM}$, store pitching-moment coefficient due to 1st empennage
CLNEM1	$(C_n)_{EM}$, store yawing-moment coefficient due to 1st empennage

The forces and moments in this common are those due to the first empennage including the lift carry over on the body, but less the contribution the body alone makes in the region of interference. All forces and moments act in the sense displayed in Figure 17 of Volume II. All moments are about the store moment center.

AD-A099 332 NIELSEN ENGINEERING AND RESEARCH INC MOUNTAIN VIEW CA F/G 20/4
PREDICTION OF SUPERSONIC STORE SEPARATION CHARACTERISTICS INCLU--ETC(U)
NOV 80 J MULLEN, F K GOODWIN, M F DILLENIUS F33615-76-C-3077
UNCLASSIFIED NEAR-TR-210-VOL-4 AFWAL-TR-80-3032-VOL-4 NL

4 OF 4
AD
A099332

END
DATE
FILMED
8-81
DTIC

COMMON /EFORC2/ CYEM2,CNEM2,CLLEM2,[illegible]MEM2,CLNEM2

CYEM2	$(C_Y)_{EM}$, store [illegible]rce coefficient due to 2nd empennage
CNEM2	$(C_N)_{EM}$, store normal-force coefficient due to 2nd empennage
CLLEM2	$(C_\ell)_{EM}$, store rolling-moment coefficient due to 2nd empennage
CLMEM2	$(C_m)_{EM}$, store pitching-moment coefficient due to 2nd empennage
CLNEM2	$(C_n)_{EM}$, store yawing-moment coefficient due to 2nd empennage
	The forces and moments in this common are those due to the second empennage including the lift carry over on the body, but less the contribution the body alone makes in the region of interference. All forces and moments act in the sense displayed in Figure 17 of Volume II. All moments are about the store moment center.

COMMON /EMPCON/ RFIN(11),YTAIL(11,4),ZTAIL(11,4),FROLE(4),FCONA,
FCONB,FCONC,CCL3(11),CCL5(11),XTAILI,DYFIN

RFIN(K)	$a + (\frac{K - 1}{MSF - 1})(s_h - a)$ or $a + (\frac{K - 1}{MSF - 1})(s_v - a)$; Figure 10, Reference 6, with $s_h = s_v$
YTAIL(K,J)	y coordinate of the Kth control point on the Jth fin
ZTAIL(K,J)	z coordinate of the Kth control point on the Jth fin
FROLE(K)	$\phi_f + 90°$ for fin 1, $\phi_f + 270°$ for fin 2, $\phi_f + 180°$ for fin 3, ϕ_f for fin 4; radians; see Figure 10, Reference 6
FCONA	$(dC_L/d\alpha)_H/ (s_h - a)^2$
FCONB	$FCONA/\ell_R$
FCONC	$\ell_f - x_{s,m}$; Figure 8, Reference 6
CCL3(K)	$(c c_\ell)_3$ and $(c c_\ell)_4$, Equations (I-14) and (I-19), Reference 6; equal since $s_h = s_v$
CCL5(K)	$(c c_\ell)_5$ if planar empennage, Equation (I-29), Reference 6; $(c c_\ell)_6$ if cruciform empennage, Equations (I-38) and (I-51), Reference 6

NASHK	maximum locations in blank common required in BSHOCK
NAFLD	maximum locations in blank common required in FLDVEL
IA0	offset location in blank common of all above arrays when multiple configurations are simultaneously in core
ID0	offset location in blank common of ID array containing /DIMENS/ information for additional source panel configurations in core
ISK0	offset location in blank common of array containing information in common /BSHOCK/ for additional source panel configurations in core
IYIM,IZIM	locations in blank common of start of image store Y and Z control point coordinates
ISVN,ISKP	locations in blank common of start of temporary arrays, SVN and LSKP
NRING	number of rings of panels on body
IROW(I)	number of panels around Ith ring of panels

COMMON /EFORC1/ CYEM1,CNEM1,CLLEM1,CLMEM1,CLNEM1

CYEM1	$(C_Y)_{EM}$, store side-force coefficient due to 1st empennage
CNEM1	$(C_N)_{EM}$, store normal-force coefficient due to 1st empennage
CLLEM1	$(C_\ell)_{EM}$, store rolling-moment coefficient due to 1st empennage
CLMEM1	$(C_m)_{EM}$, store pitching-moment coefficient due to 1st empennage
CLNEM1	$(C_n)_{EM}$, store yawing-moment coefficient due to 1st empennage

The forces and moments in this common are those due to the first empennage including the lift carry over on the body, but less the contribution the body alone makes in the region of interference. All forces and moments act in the sense displayed in Figure 17 of Volume II. All moments are about the store moment center.

XTAILI	not used
DYFIN	spanwise distance between control points on a fin

COMMON /EMPDAT/ FINSS,RADAV,XTAIL,PHIROL,MSF,IPLNR,CLALPH

FINSS	tail fin semispan; input item 15
RADAV	average body radius in fin region; input item 15
XTAIL	input item 15
PHIROL	ϕ_f; input item 15
MSF	input item 14
IPLNR	empennage type; input item 14
CLALPH	input item 15

COMMON /FLOW/ ALFACR,GAMF,FMACH,RHO,VINF,BETA,BETASQ,FMCHSQ

ALFACR	α_f, radians
GAMF	fuselage flight path angle, γ_f; input item 3
FMACH	M_∞; input item 3, Program I
RHO	ρ_∞; input item 3
VINF	V_∞; input item 3
BETA	$\beta = \sqrt{M_\infty^2-1}$
BETASQ	β^2
FMCHSQ	M_∞^2

COMMON /FORM/ CN,CT,CM,CYB,CLROLL,CNYAW,CL,CD,DXN

CN	elliptic body normal-force coefficient
CT	elliptic body axial-force coefficient
CM	elliptic body pitching-moment coefficient
CYB	elliptic body side-force coefficient
CLROLL	elliptic body rolling-moment coefficient

CNYAW	elliptic body yawing-moment coefficient
CL	lift coefficient perpendicular to free stream velocity
CD	drag coefficient parallel to free stream velocity
DXN	axial location from elliptic store nose of center of lift in normal direction

COMMON /FSHOCK/ NFSHK,FXSHK(50),FRSHK(50),FDRDX(101)

NFSHK	number of x and r values in fuselage nose shock table generated from line singularities
FXSHK(I), FRSHK(I)	x and r coordinates of Ith circular fuselage nose shock location at zero angle of attack
FDRDX(J)	dr/dx at Jth circular fuselage control point

COMMON /FSOR/ FXL(101),FSS(100),FDS(100),NFSOR

FXL(I)	array containing the x positions of the fuselage sources and doublets; positive, measured from tip of nose
FSS(I)	array containing the strengths of the circular fuselage source distribution
FDS(I)	array containing the strengths of the circular fuselage doublet distribution
NFSOR	number of circular fuselage sources and doublets; input item 9, Program I

COMMON /GRIDCU/ XPT(500),YPT(100),ZPT(100),SPHI(40),CPHI(40),
SWP(500),SNBP(40),CSBP(40),THTBP(40),DPNET(500)

XPT(I)	x_w coordinate of Ith wing/body/pylon constant u-velocity corner point
YPT(J)	y_w coordinate of Jth chordwise row of wing/body/pylon constant u-velocity corner points
ZPT(J)	z_w coordinate of Jth chordwise row of wing/body/pylon constant u-velocity corner points

SPHI(J)	sine of dihedral angle associated with Jth row of wing constant u-velocity corner points
CPHI(J)	cosine of dihedral angle associated with Jth row of wing constant u-velocity corner points
SWP(I)	leading-edge slope of semi-infinite influencing triangle associated with Ith constant u-velocity corner points
SNBP(J)	sine of orientation angle of Jth row of body u-velocity corner points
CSBP(J)	cosine of orientation angle of Jth row of body u-velocity corner points
THTBP(J)	polar angle in cross-sectional plane defining the Jth row of fuselage constant u-velocity corner points; positive in counterclockwise rotation from positive y_B axis
DPNET(I)	net strength of Ith u-velocity corner point

```
COMMON /GRIDTH/ XPTS(1000),YPTS(100),ZPTS(100),SPHS(40),CPHS(40),
     SWPS(1000),THTNET(1000),DZDX(400)
```

XPTS(I)	x_W coordinate of Ith wing/pylon thickness source panel corner point
YPTS(J)	y_W coordinate of Jth chordwise row of wing/pylon thickness source panel corner points
ZPTS(J)	z_W coordinate of Jth chordwise row of wing/pylon thickness source panel corner points
SPHS(J)	sine of dihedral angle associated with Jth row of wing source panel corner points
CPHS(J)	cosine of dihedral angle associated with Jth row of wing source panel corner points
SWPS(I)	leading-edge slope of semi-infinite influencing triangle associated with Ith source panel corner point
THTNET(I)	net strength of Ith thickness source panel corner point
DZDX(I)	dz/dx of Ith thickness panel

COMMON /HEAD/ TITLE1(20),TITLE2(20)

TITLE1	array containing hollerith description of non-circular body external geometry
TITLE2	array containing hollerith description of non-circular body paneling distribution

COMMON /IFORCE/ NDAMP, NEJSTR,NEMP,NGAM,NSEG,NHSEGO,NROLL

NDAMP	input item 4
NEJSTR	subscript associated with separated store; $1 \leq$ NEJSTR $\leq$ NSTRS
NEMP	input item 4
NGAM	input item 4
NSEG	input item 4
NHSEGO	not used
NROLL	input item 4

COMMON /INTRDT/ PHIDIH,THETIT,PHIFR,PHIFL,PHIFU,PHIFD

PHIDIH	interdigitated tail dihedral angle; input item 21
THETIT	interdigitated tail meridian attachment angle; input item 21
PHIFR	dihedral angle of fin 1; input item 22
PHIFL	dihedral angle of fin 2; input item 22
PHIFU	dihedral angle of fin 3; input item 22
PHIFD	dihedral angle of fin 4; input item 22

COMMON /JECTOR/ NJECTR,FXS,FYS,FZS,RMX,RMY,RMZ

NJECTR	ejector force indicator; input item 4 NJECTR=1, read ejector force input NJECTR=2, initialize ejector forces NJECTR=3, compute ejector force versus time or distance Value is incremented internally

FXS,FYS,FZS	components of ejector force along x_s,y_s,z_s axes; pounds
RMX,RMY,RMZ	components of ejector moments acting about the x_s,y_s,z_s axes and about the moment center; foot-pounds

COMMON /LETEM/ FMXT3,FMXT4

FMXT3	leading-edge Mach number associated with wing thickness at the y_w,z_w location of the store center of moments
FMXT4	trailing-edge Mach number associated with wing thickness at the y_w,z_w location of the store center of moments

COMMON /LOCBET/ FBETAL,RBETAL,SBETAL(7)

FBETAL	value of β to be used in calculating the fuselage induced velocities at a specific field point
RBETAL	value of β to be used in calculating the rack induced velocities at a specific field point
SBETAL(J)	value of β to be used in calculating the Jth store induced velocities at a specific field point

COMMON /MULSTR/ NESHPT,NEJSHP,NEJGB,LASTEJ,LASTF,LASTA,
IDEJ,IDF,ID2

NESHPT	number of elliptic store shapes
NEJSHP	index of separated store shape
NEJGB	for separated store shape, index of separated store
LASTEJ	last location in blank common of separated elliptic store solution arrays
LASTF	last location in blank common of noncircular fuselage solution arrays
LASTA	last location in blank common of second elliptic store shape solution arrays

IDEJ	index locating index array, ID, of separated store shape
IDF	index locating index array, ID, of noncircular fuselage
ID2	index locating index array, ID, of second elliptic store shape

COMMON /NEWFOR/ NTHETA,DTHETA,THETAD(37),STHETA(37),CTHETA(37),
REFL,SREF,NCONPT

NTHETA	number of circumferential points to be used in circular store force and moment calculation; input item 4
DTHETA	$\Delta\theta=2\pi(\text{NTHETA}-1)$
THETAD(J)	$\theta_J=(J-1)\Delta\theta$, deg.
STHETA(J)	$\sin\theta_J$
CTHETA(J)	$\cos\theta_J$
REFL	reference length used in force and moment calculation
SREF	reference area used in force and moment calculation
NCONPT	number of control points used in circular store source and doublet strength calculation

COMMON /NUFLOW/ FNUMCH,BETANU,BTSQNU,INUMCH,ISMTH,XT3,XT4

FNUMCH	local Mach number, M_ℓ, to be used in wing thickness and wing-fuselage u-velocity panel velocity calculations at a specific field point
BETANU	$\sqrt{M_\ell^2-1}$
BTSQNU	M_ℓ^2-1

INUMCH — INUMCH=1 when calculating wing-fuselage u-velocity panel and wing thickness velocities; INUMCH=0 otherwise

ISMTH — not used

XT3 — x_w coordinate of wing leading-edge shock at y_w, z_w coordinates of store center of moments

XT4 — x_w coordinate of wing trailing-edge expansion cone at y_w, z_w coordinates of store center of moments

COMMON /NUINDX/ NRW,NRP,NRB,NPTW,NRWS,NRPS,NPTWS,NRWP,NPTWP,NCW1,
NCP1,NCWB1,NCWS,NCWS1,NCPS1

NRW — number of rows of corner points defined for wing constant u-velocity panels; NRW=MSW+1+KW; MSW is input, item 36, Program I

NRP — number of rows of corner points defined for pylon constant u-velocity panels; NRP=MSP+1+KP; MSP is input, item 44, Program I

NRB — number of rows of corner points defined for fuselage constant u-velocity panels; NRB=2*(NBDCR1+NBDCR2); NBDCR1 and NBDCR2 are input, item 9, Program I

NPTW — number of corner points defined for wing constant u-velocity panels; NPTW=NCW1*NRW

NRWS — number of rows of corner points defined for wing source panels; NRWS=MSWS+1+KW; MSWS is input, item 40, Program I

NRPS — number of rows of corner points defined for pylon source panels; NRPS=MSPS+1+KP; MSPS is input, item 46, Program I

NPTWS — number of corner points defined for wing source panels; NPTWS=NCWS1*NRWS

NRWP — NRW+NRP

NPTWP	number of corner points defined for wing and pylon constant u-velocity panels; NPTWP= NPTW+NCP1*NRP
NCW1	number of corners in a chordwise row of wing constant u-velocity panels; NCW+1; NCW is input, item 36, Program I
NCP1	number of corners in a chordwise row of pylon constant u-velocity panels; NCP+1; NCP is input, item 44, Program I
NCWB1	number of corners in a chordwise row of fuselage constant u-velocity panels; NCWB+1; NCWB is input, item 9, Program I
NCWS	number of wing source panels in a chordwise row; input item 40, Program I
NCWS1	number of corners in a chordwise row of wing source panels; NCWS+1; NCWS in input, item 40, Program I
NCPS1	number of corners in a chordwise row of pylon source panels; NCPS+1; NCPS is input, item 46, Program I

COMMON /OAFM/ XM,ZM,CZOA,CYOA,CMOA,CLNOA,CLLOA

XM	X-location of moment center relative to store nose, positive aft
ZM	Z-location of moment center relative to store centerline, positive up
CZOA	$(C_N)_E$, total empennage normal-force coefficient
CYOZ	$(C_Y)_E$, total empennage side-force coefficient
CMOA	$(C_m)_E$, total empennage pitching-moment coefficient
CLNOA	$(C_n)_E$, total empennage yawing-moment coefficient

CLLOA | $(C_\ell)_E$, total empennage rolling-moment coefficient

The forces and moments in this common are those due to the sum of the contributions of each of the fins present and the body interference shell in the presence of both the store body and its reflections and the parent aircraft. All forces and moments act in the sense displayed in Figure 17 of Volume II. All moments are about the store moment center

```
COMMON /ONE/ DELTP(250),FN(250),PNLC(250),SWPPLE(250),SWPPTE(250),
     XBAR(250),ZBAR(250),XCPT(250),YCPT(250),ZCPT(250),XLF(250),
     XLB(250),XRF(250),XRB(250),YLC(250),YRC(250),ZLF(250),
     ZRF(250),ZLB(250),ZRB(250),SNT(100),CST(100),SNT2(100),
     CST2(250),XFBIP(100),ALFA,ALFR,B2,B2V,BETA,BETAR,CONST,
     CN,DX,EM,FMACH,SINALF,SINBET,SLOPE,TLRNC,TOTLR,U,V,W,
     UCHK,VCHK,WCHK,WBIP,X,Y,Z,IV,IF,II,JV,MSWR,MSWL,MSLU,
     MSWD,NBIP,NCW,NHP,NPR,NRP,N3P,NOCPT,NOUT,NPANLS,NWBP,
     RA,RB,ERATIO,BODY,DELTA
```

DELTP(J)	Δcp_{lin}, linear loading pressure coefficient of Jth u-velocity panel
FN(J)	normal force divided by q for Jth u-velocity panel
PNLC(J)	panel chord through control point of Jth u-velocity panel
SWPPLE(J)	dx/dy, leading edge slope of Jth u-velocity panel measured from local panel y-axis
SWPPTE(J)	dx/dy, tangent of trailing edge sweep angle of Jth u-velocity panel measured from local y-axis
XBAR(J)	x-location of centroid of Jth u-velocity panel in wing coordinate system
ZBAR(J)	z-location of centroid of Jth u-velocity panel in wing coordinate system
XCPT(J), YCPT(J), ZCPT(J)	coordinates of control point of Jth u-velocity panel in wing coordinate system
XLF(J)	x-location of left front corner of Jth u-velocity panel in wing coordinate system (corner 1, Figure 3, Volume II)

XLB(J)	x-location of left back corner of Jth u-velocity panel in wing coordinate system (corner 3, Figure 3, Volume II)
XRF(J)	x-location of right front corner of Jth u-velocity panel in wing coordinate system (corner 2, Figure 3, Volume II)
XRB(J)	x-location of right back corner of Jth u-velocity panel in wing coordinate system (corner 4, Figure 3, Volume II)
YLC(J)	y-location of left side edge of Jth u-velocity panel in wing coordinate system
YRC(J)	y-location of right side edge of Jth u-velocity panel in wing coordinate system
ZLF(J)	z-location of left front corner of Jth u-velocity panel in wing coordinate system (corner 1, Figure 3, Volume II)
ZRF(J)	z-location of right front corner of Jth u-velocity panel in wing coordinate system (corner 2, Figure 3, Volume II)
ZLB(J)	z-location of left back corner of Jth u-velocity panel in wing coordinate system (corner 3, Figure 3, Volume II)
ZRB(J)	z-location of right back corner of Jth u-velocity panel in wing coordinate system (corner 4, Figure 3, Volume II)
SNT(J)	sin(THTI(J)), $\sin\theta_{BIP,j}$ in Figure 18 of Volume II, locates panel left side edge for Jth body interference shell u-velocity panel
CST(J)	cos(THTI(J)),$\cos\theta_{BIP,j}$ in Figure 18 of Volume II, locates panel left side edge for Jth body interference shell u-velocity panel
SNT2(J)	$\sin\theta_{2,BIP}$ in Figure 18 of Volume II, used in panel rotation transformation for Jth body interference shell u-velocity panel
CST2(J)	$\cos\theta_{2,BIP}$ in Figure 18 of Volume II, used in panel rotation transformation for Jth body interference shell u-velocity panel
XFBIP(J)	x-location of leading edge of Jth body interference shell u-velocity panel

ALFA	angle of pitch, deg.
ALFR	angle of pitch, radians
B2	exposed horizontal fin semispan; input item 19
B2V	exposed vertical fin semispan; input item 20
BETA	$\beta = \sqrt{M_\infty^2-1}$
BETAR	angle of side slip, radians
CONST	4π
CN	not used
DX	body interference u-velocity panel length
EM	local u-velocity panel leading or trailing edge slope
FMACH	M_∞, free stream Mach number
SINALF	sin(ALFR)
SINBET	sin(BETAR)
SLOPE	local u-velocity panel leading or trailing edge slope
TLRNC	numerical tolerance related to semispan
TOTLR	2*TLRNC
U,V,W	perturbation velocities induced by a single corner of a constant u-velocity panel in local panel coordinates
UCHK,VCHK,WCHK	perturbation velocities induced at a point by a specified number of constant u-velocity panels in wing coordinates
WBIP	width of body interference panel
X,Y,Z	coordinates of a field point relative to a corner of a constant u-velocity panel aligned with that panel
IV	index of influencing u-velocity panel
IF	final index in loop over number of influencing panels
II	initial index in loop over number of influencing panels

JV	index of u-velocity panel control point
MSWR	number of panels in spanwise direction on right fin; input item 18
MSWL	number of panels in spanwise direction on left fin; input item 18
MSWU	number of panels in spanwise direction on upper fin; input item 18
MSWD	number of panels in spanwise direction on lower fin; input item 18
NBIP	number of body interference panels; =NBDCR*NCWB
NCW	number of panels in chordwise direction; input item 18
NHP	number of horizontal fin panels (fins 1 and 2); =NCW*(MSWR+MSWL)
NPR	print control index
NPR	number of right fin (fin 1) panels; =NCW(MSWR)
N3P	number of right, left, and upper fin panels (fins 1, 2 and 3); =NCW(MSWR+MSWL+MSWU)
NOCPT	control index signifying whether or not a u-velocity panel control point is under consideration
NOUT	print control: 1=yes, 0=no
NPANLS	total number of u-velocity panels on fins; =NCW*(MSWR+MSWL+MSWU+MSWD)
NWBP	total number of u-velocity panels on fins; =NPANLS+NBIP
RA	vertical semi-axis of elliptic store at interference shell
RB	horizontal semi-axis of elliptic store at interference shell
ERATIO	elliptic ratio (RB/RA)
BODY	logical indicator of interference shell presence (=NBDCR.NE.0)

DELTA	logical indicator of presence of fin deflection

COMMON /OUTINI/ XNOSEI,YNOSEI,ZNOSEI,XCGI,YCGI,ZCGI,XBASEI,
YBASEI,ZBASEI

XNOSEI	ξ coordinate of tip of separated store nose at t = 0
YNOSEI	η coordinate of tip of separated store nose at t = 0
ZNOSEI	ζ coordinate of tip of separated store nose at t = 0
XCGI	ξ coordinate of separated store moment center at t = 0
YCGI	η coordinate of separated store moment center at t = 0
ZCGI	ζ coordinate of separated store moment center at t = 0
XBASEI	ξ coordinate of separated store base at t = 0
YBASEI	η coordinate of separated store base at t = 0
ZBASEI	ζ coordinate of separated store base at t = 0

COMMON /PARAM/ XMACH,ALPHA,BETA,ALPHAC,PHIR,EM,SINAC,COSAC,
SPHI,CPHI,SINA,SINB

XMACH	Mach number seen by source panels
ALPHA	α, free stream angle of attack seen by source panels, degs. $=\sin^{-1}(\sin(\text{ALPHAC})\cdot\cos(\text{PHIR}))$
BETA	β, free stream angle of side slip seen by source panels, degs. $=\sin^{-1}(\sin(\text{ALPHAC})\cdot\sin(\text{PHIR}))$
ALPHAC	α_c, included angle of attack seen by source panels, degs.
PHIR	ϕ_R, angle of roll seen by source panels
EM	temporary Mach number of last computation
SINAC	$\sin(\alpha_c)$

COSAC	$\cos(\alpha_c)$
SPHI	$\sin(\phi_r)$
CPHI	$\cos(\phi_r)$
SINA	$\sin(\alpha)$
SINB	$\sin(\beta)$

```
COMMON /PYGEOM/ Z(20),XPLE,YPL,CRP,HP,PSIPLE(20),PSIPTE(20),IP,
     SLLE,PSLPDF,CENTER,ZPL,LVSPP
```

Z(K)	input item 45, Program I
XPLE	input item 43, Program I
YPL	y_w location of the pylon; YPL=Y(IP) of input item 37, Program I
CRP	input item 43, Program I
HP	input item 43, Program I
PSIPLE(K)	input item 45, Program I
PHIPTE(K)	input item 45, Program I
IP	input item 43, Program I
SLLE	slope of pylon leading edge
PSLPDF	difference in leading and trailing edge slopes, PSLPDE=SLLE-SLTE
CENTER	logical test for fuselage centerline pylon (CENTER=YPL.EQ.0)
ZPL	z_w location of the pylon; ZPL=ZLC((IP-2)NCW+1)
LVSPP	breaks in pylon sweep indicator; LVSPP=0, no; LVSPP=1, yes

```
COMMON /RKGEOM/ RRMAX,RLTHC,XWROC,YWRO,ZWRO,NRPOLY,RXEND(7),
     RCOEF(7,7),XBRO,YBRO,ZBRO,RIBCR,SRIBCR,CRIBCR
```

RRMAX	maximum rack radius; input item 49, Program I
RLTHC	length of rack; input item 49, Program I

XWROC	x_W location of rack in wing coordinates, feet
YWRO	y_W location of rack in wing coordinates, feet
ZWRO	z_W location of rack in wing coordinates, feet
NRPOLY	number of rack shape polynomials; input item 50, Program I
RXEND(J)	rack polynomial end points; input item 51, Program I
RCOEF(I,J)	rack polynomial coefficients; input item 52, Program I
XBRO	x_B location of rack in body coordinates, feet
YBRO	y_B location of rack in body coordinates, feet
ZBRO	z_B location of rack in body coordinates, feet
RIBCR	RIC·DTOR; input item 49 for RIC in Program I
SRIBCR	sin(RIBCR)
CRIBCR	cos(RIBCR)

COMMON /RSHOCK/ NRSHK,RXSHK(50),RRSHK(50),RDRDX(101)

NRSHK	number of x and r values in rack nose shock table generated from line singularities
RXSHK(I), RRSHK(I)	x and r coordinates of Ith rack nose shock location at zero angle of attack, feet
RDRDX(J)	dr/dx at Jth rack control point

COMMON /RSOR/ RXL(101),RSS(100),RDS(100),NRSOR

RXL	array containing the x positions of the rack sources; positive, measured from tip of rack nose
RSS	array containing the strengths of the rack source distribution
RDS	array containing the strengths of the rack doublet distribution
NRSOR	number of rack sources and doublets; input item 53, Program I

COMMON /SORDUB/ BETAA,SRS(150),VRATSA,BETAS,BSS,DSW(150),DSV(150),
XS(150),XD(150),BETAAI,BETASI,BSSI

BETAA	$\beta_a = \sqrt{M_a^2-1}$ where M_a is the axial Mach number seen by separated store
SRS(J)	strength of the Jth source modeling the separated store
VRATSA	total free-stream velocity seen by the separated store divided by the axial component
BETAS	$\beta_s = \sqrt{M_s^2-1}$ where M_s is the total Mach number seen by the separated store; $M_s = M_\infty V_{\infty s}/V_\infty$
BSS	β_s^2
DSW(J)	strength of the Jth upwash doublet modeling the separated store
DSV(J)	strength of the Jth sidewash doublet modeling the separated store
XS(J)	origin of the Jth source modeling the separated store
XD(J)	origin of the Jth upwash and sidewash doublets modeling the separated store
BETAAI	not used
BETASI	not used
BSSI	not used

COMMON /SPCPRS/ DLTP(250)

DLTP(J)	ΔC_p, Bernoulli pressure used in loading calculation for Jth u-velocity panel on empennage; ΔC_p is computed in Equation (10), Reference 6

COMMON /SPSANG/ SINALC,COSALC,SINPHI,COSPHI

SINALC	$\sin(\alpha_c)$
COSALC	$\cos(\alpha_c)$

SINPHI $\sin(\phi_r)$

COSPHI $\cos(\phi_r)$

COMMON /SSHOCK/ NSSHK(7),SXSHK(50,7),SRSHK(50,7),SDRDX(101,7)

NSSHK(J) number of x and r values in Jth store nose shock table generated from line singularities

SXSHK(I,J), SRSHK(I,J) x and r coordinates of Ith circular store shock location at zero angle of attack for Jth store

SDRDX(K,J) dr/dx at Kth circular store control point of Jth store

COMMON /SSOR/ SXL(101,7),SSS(100,7),SDS(100,7),NSSOR(7)

SXL(I,J) x-position of Ith source for Jth circular store; positive measured from tip of nose

SSS(I,J) strength of Ith source for Jth circular store

SDS(I,J) strength of Ith doublet for Jth circular store

NSSOR(J) number of Jth circular store sources and doublets; equal to input item 56, Program I

COMMON /STGEOM/ SLTHC(7),SRMAX(7),SIBCR(7),CSIBCR(7),SSIBCR(7),
YBSO(7),XBSO(7),ZBSO(7),NUMSTR(7),NSHAPE(7),NSHPT,
MSHAPE(7),SPHIRR(7)

SLTHC(J) length of Jth store; input item 54, Program I

SRMAX(J) maximum radius of Jth store; input item 54, Program I

SIBCR(J) SIC(J)•DTOR; input item 54, Program I, for SIC(J)

SSIBCR(J) sin(SIBCR(J))

CSIBCR(J) cos(SIBCR(J))

XBSO(J) x_B coordinate of tip of nose of Jth store

YBSO(J) y_B coordinate of centerline of Jth store

ZBSO(J) z_B coordinate of tip of nose of Jth store

NUMSTR(J) store number; input item 54, Program I

NSHAPE(J)	store shape number, input item 54, Program I
NSHPT	number of different store shapes; input item 55, Program I
MSHAPE(K)	shape number of Kth store shape; input item 56, Program I
SPHIRR(J)	initial roll angle of Jth store coordinate axes; positive right wing down, radians

COMMON /STORIC/ BTNOSC,RSSC

BTNOSC	β associated with first source of image store on fuselage centerline (=XSR/RSSC)
RSSC	distance from real store nose to fuselage centerline; locates line source image on fuselage centerline. See Figure C-7.

COMMON /STORIF/ IMFSTR,BTNOSF,RSS,YBN,ZBN,SFAC

IMFSTR	index indicating whether store shock reflecting from fuselage strikes the store (IMFSTR=0, no; IMFSTR=1, yes)
BTNOSF	β associated with first source and doublet of fuselage image store (=XSR/RSS). See Figure C-7.
RSS	location of image store; RSS = RSSC-RBS^2/RSSC
YBN,ZBN	Y_B,Z_B location of store nose in fuselage coordinates
SFAC	image line doublet multiplying factor (=RBS^2/RSSC^2)

COMMON /STORIM/ IMSTOR,BTNOSE,XWN,YWN,ZWN,IMAGE

IMSTOR	index indicating whether store shock reflecting from wing strikes the store (IMSTOR=0, no; IMSTOR=1, yes)
BTNOSE	β associated with first source and doublet of wing image store
XWN,YWN,ZWN	x,y,z coordinates of wing image store nose in wing coordinate system

IMAGE	logical index indicating whether image store is present (IMAGE=T, yes; IMAGE=F, no)

```
COMMON /STORSD/ NXBODY,XBD(151),RBD(151),NPOLY,XEND(7),COEF(7,7),
     XBC(150),RBC(150),DRDXBC(150),NSORCE,DRDXBD(151)
```

NXBODY	number of circular store body definition points (=NSORCE+1)
XBD(J), RBD(J)	x and r coordinates of circular store body definition points at Jth station
NPOLY	number of shape polynomials, input item 4
XEND(J)	end points of polynomials, input item 11
COEF(J,K)	coefficients of polynomials, input item 12
XBC(J), RBC(J)	x and r coordinates of circular store body control points for Jth station
DRDXBC(J)	dr/dx, slope of body radius distribution of circular store at Jth control point
NSORCE	number of sources and doublets, input item 4
DRDXBD(J)	dr/dx, slope of body radius distribution of circular store at Jth body definition station

```
COMMON /STRBET/ BTAL,BTSL
```

BTAL	local value of β used in calculating wing image store line source induced velocities
BTSL	local value of β used in calculating wing image store line doublet induced velocities

```
COMMON /STRBTC/ BTALC,BTSLC
```

BTALC	local value of β used in calculating fuselage centerline image store line source induced velocities
BTSLC	local value of β used in calculating fuselage centerline image store line doublet induced velocities

COMMON /STRBTF/ BTALF,BTSLF

BTALF	local value of β used in calculating fuselage image store line source induced velocities
BTSLF	local value of β used in calculating fuselage image store line doublet induced velocities

COMMON /SWEEPS/ VSWLER(20),VSWTER(20),VSWLEL(20),VSWTEL(20),
VSWLEU(20),VSWTEU(20),VSWLED(20),VSWTED(20),LVSWP,LEFT,
FAC,NCWB,ARPNL(250),WIDTH(250)

VSWLER(J)	right fin optional leading-edge sweep; input item 25
VSWTER(J)	right fin optional trailing-edge sweep; input item 25
VSWLEL(J)	left fin optional leading-edge sweep; input item 26
VSWTEL(J)	left fin optional trailing-edge sweep; input item 26
VSWLEU(J)	fin 3 optional leading-edge sweep; input item 27
VSWTEU(J)	fin 3 optional trailing-edge sweep; input item 27
VSWLED(J)	fin 4 optional leading-edge sweep; input item 28
VSWTED(J)	fin 4 optional trailing-edge sweep; input item 28
LVSWP	spanwise breaks in wing sweep option; input item 18
LEFT	logical index indicating geometry is laid out on left side of centerline
FAC	fraction of local panel chord through control point of u-velocity panel at which velocity influence is computed
NCWB	number of panels in axial direction on body interference shell; input item 18
ARPNL(J)	surface area of Jth u-velocity panel
WIDTH(J)	width of Jth u-velocity panel

COMMON /TEVRT/ GAMTE(20),YCG(20),ZCG(20)

GAMTE(I)	Γ_I/V_∞, net vorticity at trailing edge of store empennage generated from spanload distributions on fins for Ith vortex. See Appendix B, Reference 5
YCG(I),ZCG(I)	y_s and z_s location in crossflow plane of Ith trailing edge vortex

COMMON /THREE/ ANGLR,ANGLL,ANGLU,ANGLD,DELR,DELL,DELU,DELD,SREF, REFL

ANGLR	$\alpha + \delta_R$
ANGLL	$\alpha + \delta_L$
ANGLU	$\beta + \delta_U$
ANGLD	$\beta + \delta_D$
	The above four variables are valid only for cruciform fin or monoplane wing configurations; radians
DELR	δ_R, fin 1 deflection; input item 21
DELL	δ_L, fin 2 deflection; input item 21
DELU	δ_U, fin 3 deflection; input item 21
DELD	δ_D, fin 4 deflection; input item 21
SREF	empennage reference area
REFL	empennage reference length

COMMON /THRUST/ NTPOLY,TEND(5),TC(5,6),FTHRUS,NTHRUS

NTPOLY	number of thrust polynomials; input item 29
TEND(I)	time end points of Ith thrust polynomial; input item 30
TC(I,J)	Jth thrust polynomial coefficient of Ith polynomial; input item 31
FTHRUS	thrust force, pounds
NTHRUS	thrust option index; input item 4

COMMON /TOTFOR/ CSIDE,CNORM,CROLL,CPITCH,CYAW

CSIDE	store total side-force coefficient
CNORM	store total normal-force coefficient
CROLL	store total rolling-moment coefficient
CPITCH	store total pitching-moment coefficient
CYAW	store total yawing-moment coefficient
	The forces and moments in this common represent the total values acting on the store not including any thrust or ejector forces and moments. All forces and moments act in the sense displayed in Figure 17 of Volume II. All moments are about the store moment center.

COMMON /TRMSSD/ ARG,ROOT,ACOSH

ARG	argument of line source and doublet expressions
ROOT	square root used in line source and doublet expressions
ACOSH	inverse cosh quantity used in line source and doublet expressions

COMMON /VEL/ VXAV,WXAV,USTOR,NOUT,NV,NS,NF

VXAV,WXAV	average crossflow velocity components, v and w, at axial station, x, along elliptic store body centerline due to presence of parent aircraft
USTOR	$v_x/V_{\infty s}$, axial free stream velocity component of store
NOUT	print control; set to zero to eliminate print
NV	number of vortices
NS	not used
NF	not used

COMMON /VELARG/ X,Y,Z,U,V,W,EM,TLRNC,PYPNL,UP,VP,WP

X,Y,Z	x,y,z coordinates of field point in semi-infinite triangle coordinate system
U,V,W	u,v,w influence functions due to one semi-infinite triangle
EM	leading-edge slope of semi-infinite triangle
TLRNC	numerical tolerance based on wing semispan
PYPNL	logical variable indicating a pylon semi-infinite triagle; TRUE=pylon triangle, FALSE=no pylon triangle
UP,VP,WP	sum of velocities at a field point due to wing thickness, pylon thickness, wing u-velocity panels, pylon u-velocity panels, or fuselage interference shell

COMMON /VRTXV/ VVRTX(250),WVRTX(250),NVRTIN,NVRTX,VRTMAX

VVRTX(J), WVRTX(J)	v and w components of velocity at the Jth empennage control point generated due to the presence of external vortices
NVRTIN	vorticity calculation override index; input item 24
NVRTX	number of vortices influencing empennage u-velocity panels
VRTMAX	maximum vortex induced velocity; input item 23

COMMON /WBTR/ THTI(100),XWLE

THTI(J)	meridian angle used in defining Jth u-velocity panel edge on store interference shell; degs.
XWLE	axial distance measured from store nose of leading edge of root chord for empennage

COMMON /WDY1/ VAV(50),WAV(50),XAV(50),NR

VAV(J)	average lateral velocity, v, at Jth ring of store body source panels
WAV(J)	average vertical velocity, w, at Jth ring of store body source panels
XAV(J)	average axial control point location of Jth ring of store body source panels
NR	number of rings for which average velocities are computed

COMMON /WGEOM/ XBWOC,ZBWO,CRW,ZDIHED

XBWOC	x_B coordinate of wing root chord leading edge; input item 13, Program I
ZBWO	z_B coordinate of wing root chord leading edge; input item 13, Program I
CRW	wing root chord; input item 34, Program I
ZDIHED	logical variable indicating whether or not there is wing dihedral; ZDIHED=TRUE, no dihedral; =FALSE, there is dihedral

COMMON /XSHOLD/ FXSHLD,RXSHLD,SXSHLD(7)

VXSHLD	x_B location of circular fuselage shoulder, feet
RXSHLD	x_R location of rack shoulder, feet
SXSHLD(J)	x_S location of Jth circular store shoulder, feet

D-3 Blank Common

The requirement for handling the many large arrays associated with the solutions for empennage constant u-velocity and body source panel strengths has necessitated both the use of out of core data handling and storage and the setting aside of a scratch storage area in core to be used for more than one purpose. To handle the latter data requirement blank common has been reserved for all calculations involving large arrays. The program flow of calculations is thus arranged to allow variables to be read from or written to external files as needed. The following describes the Program II sequence of references to external files and which arrays reside in blank common at each point in the program flow. The information residing in each of the external files is described later.

In Program II blank common is used for three purposes: (1) temporary storage of data during the transfer from TAPE12 to internal files; (2) dynamic allocation of the arrays with multiple noncircular body descriptions; and (3) temporary storage of the empennage constant u-velocity influence coefficient matrices. The descriptions which follow focus on the definition of quantities during these phases. When arrays for both the elliptic store and a noncircular fuselage are present, common definitions will be identified. Only the dimensions of arrays will vary between components. Emphasis will be given to the second two items since these data items are continuously being swapped in core during the trajectory integration. The flow chart in Figure C-2 of Appendix C identifies several points in the program where external files are referenced. The comments which follow are keyed to the usage of blank common at these points wherever possible. The descriptions of the use of blank common during the three phases follow.

The first use of blank common, item 1 of Figure C-2, is to store intermediate arrays when transferring noncircular fuselage and elliptical store data from TAPE12, written by Program I, to TAPE11 and TAPE10, respectively. These file transfers are made in

routine FRSTRT. The variables transferred are those detailed in the description of subroutine FRSTRT in Appendix C. The data for the noncircular fuselage, if the option is being used, is first copied into blank common in the same form as it existed in Program I and then copied onto TAPE11 for later use. The data for up to two elliptic store shapes is similarly copied onto TAPE10. See the descriptions for FRSTRT for the sequence of data transfers.

The two remaining uses of blank common are alternately interchanged during each integration step. The first of these uses of blank common is for containment of the arrays and variables associated with the separating elliptic store shape, a noncircular fuselage, and any second elliptic store shape. The variables which describe each of these configurations are dynamically allocated space in blank common during execution depending on the panel layout of each configuration. All information required to compute the solutions required by each of the configurations will reside in core simultaneously. The allocation of variable locations in blank common is assembled from the individual data for each of the three possible noncircular body shapes in routine STRDAT indicated at item 2 in Figure C-2. The data for each of the individual configurations is copied from files TAPE10 and TAPE11 as needed.

The separating elliptic store shape data are handled first. The data for the appropriate store shape are copied from TAPE10 into labeled common DIMENS, PARAM, BOPTNS, GEOMB, HEAD, and BSHOCK and any data not associated with these commons stored at the beginning of blank common. All the noncircular fuselage and second elliptic store shape data, if any, are copied directly into blank common. Both the control variables and geometric and strength arrays are copied sequentially into blank common immediately after the previous configuration data. To identify the location of the data, the first and last variable locations of this data are saved in common MULSTR for the separating store shape, the fuselage and an additional elliptic store shape. When multiple elliptic stores of the same shape are to

be used, all stores of the same shape share the same geometric and control variables with only the panel strengths for each store saved separately.

Tables D-1 and D-2 identify the locations in blank common of each of the variables for each of the noncircular shape configurations. Table D-1 specifies the locations of each of the variables in blank common used to describe the separating store. For each variable, index, or array, the name it is identified by in program usage, its length, the first location relative to the offset value of the configuration data, and address by which it can be referred are identified. The variable names given are those by which they are referred in internal program logic and in Program I. The next five arrays are set aside for values computed during the store trajectory. Where variable indices are used for the lengths of arrays, the number of locations occupied are dependent on the program input. The address of any variable is specified by an offset value and a value relative to the offset. The offset value is defined for the separating store (IDEJ), the noncircular fuselage (IDF), and a second elliptic store shape (ID2). The value relative to the offset specifies the number of common locations from the offset in which the value resides. Where variable names are specified for locations from the offset, the values are a function of program input. For the separating store, these variable locations are stored in common DIMENS. Refer to the descriptions of variables in DIMENS in Section D-2 for their definitions. The last column in Table D-1 presents a typical variable address in blank common by which the first value in each array may be referenced. The definitions of the variables for the separating store shape by their name in the second column follow:

XPT(I)	x-station of control point of Ith panel
YPT(I)	y-station of control point of Ith panel
ZPT(I)	z-station of control point of Ith [illegible]el
THET(I)	inclination angle at Ith panel control point

DELTA(I)	incidence angle of Ith panel
AREA(I)	surface area of Ith panel
XC(L,M)	panel corner points at Lth axial station of Mth segment
YC(L,N,M)	y corner point at Lth axial station of Nth meridional angle of Mth segment
ZC(L,N,M)	z corner point at Lth axial station of Nth meridional angle of Mth segment
GB(I,J)	strength of Ith panel for Jth store of given shape
U(I)	u-velocity component induced at Ith source panel
V(I)	v-velocity component induced at Ith source panel
W(I)	w-velocity component induced at Ith source panel
CP(I)	pressure coefficient at Ith source panel control point

Table D-2 specifies the locations of each of the variables in blank common used by the noncircular fuselage. For each of the variables, the name, length, first location relative to the offset value for the fuselage, IDF, and the address in blank common for referencing each variable are specified. The meanings of each of the columns are the same as for the separating store data in Table D-1. In the last column, integer and logical variables are referenced by variable names of equivalent type. An array of similar type, here indicated by IA for integer variables and LA for logical variables, are used when accessing those variable types. Such arrays of length of one and the correct type specification are equivalenced to the first value of blank common to avoid mixed mode operations.

In addition to the geometric arrays shown in Table D-1 for the separating store, all the additional variables necessary to locate them and additional shape and shock location data are also contained in blank common. The additional information previously contained in labeled commons DIMENS, PARAM, BOPTNS, GEOMB, HEAD, and BSHOCK in Program I are now in blank common. The values here are those defined for the noncircular fuselage. The definitions of variables 1 through 52 in Table D-2 are the same as those

found under DIMENS in Section D-2. The descriptions for fuselage variables 53 through 64 are found under PARAM. The variables 65 through 88 are described under labeled common BOPTNS. Variables 89 through 95 are described under labeled common GEOMB. Variables 96 and 97 are described under labeled common HEAD. Variables 98 through 110 for the fuselage are similarly described under the definitions of common BSHOCK. Lastly, variables 111 through 120 have the same definitions as the arrays defined for the separated store shape in Table D-1.

Though all values are presented relative to the offset value for the fuselage, IDF, variables 47, 48, and 49 may also be used as offset values for different segments of the variable list. Variable 48, ID0, is the same as IDF and may be used as the offset for variables 1 through 52 for the fuselage. Variable 49, ISK0, is equal to IDF+571 and is used to serve as the offset for variables 98 through 110. Similarly, variable 47, IA0, is equal to IDF+811 and is used to serve as the offset for the arrays in variables 111 through 120.

The variables for the second store shape are copied into blank common in the same order as shown for the fuselage in Table D-2 except that the value ID2 is substituted everywhere for IDF. After generation of the above blank common layout in STRDAT at the point indicated as item 2 in Figure C-2, a copy of all variables in blank common specified in Tables D-1 and D-2 is saved on external file TAPE7. A total of NTAP7 variables are written on TAPE7. Though certain variables in this list are permitted to change value during the trajectory calculations, their locations remain fixed for the duration of the program execution.

In addition to the first NTAP7 variables several temporary arrays are defined during program execution which use space in blank common. They are used to contain intermediate data which is no longer needed once the calculations have been performed.

In each case the arrays are used for calculations only in the routine in which they are defined or the subroutines which they call directly. These additional temporary arrays are defined at three points in Program II.

The first set of temporary arrays is defined in routine BDCOEF for use in computing the influence coefficient arrays for the effect of source panels on the elliptic store body at empennage panel control points. This operation is performed at the point indicated as item 3 in Figure C-2. The additional arrays are set aside immediately after the first NTAP7 variables in blank common. During these calculations blank common is equivalently dimensioned as

```
COMMON  A(NTAP7), XF(NFLD),YF(NFLD),ZF(NFLD),SVN(NFLD),
        LSKP(NFLD),UB(3,NFLD,MAXKR)
```

where the above arrays are defined to contain the following information

XF(J),YF(J), ZF(J)	coordinates of Jth empennage control point at which each source panel influence coefficient is computed. For empennage fins, points are identical to u-velocity panel control points. For interference shell, points are extrapolated to local surface of ellipse.
SVN(J), LSKP(J)	work arrays for checking vanishing influence at Jth control point
UB(I,J,K)	coefficient array to hold u,v,w velocity components for influence of Kth source panel in a ring at Jth empennage control point above

The results for the influence coefficients for a ring of panels on each of the NFLD control points is saved on external file TAPE10 after each calculation. When multiple empennages exist, the coefficients for the second empennage are saved immediately after the first.

The second set of temporary arrays is defined in routine SFORC2 for use in computing intermediate results for the loads on the separating store body. This operation is performed at the point indicated as item 5 in Figure C-2. Space for eight arrays is set aside in blank common. They are allocated space immediately after the first NTAP7 variables and overwrite the first set of temporary arrays. During these calculations blank common is equivalently dimensioned as

```
COMMON  A(NTAP7), UT(NBODY),VT(NBODY),WT(NBODY),YIM(NBODY),
        ZIM(NBODY),SVN(NBODY),LSKP(NBODY),AN(KRAD,KRAD)
```

where the above arrays are defined to contain the following information

UT(J),VT(J), WT(J)	three components of velocity in store source panel coordinate system containing the influence of the parent aircraft and free stream velocities at Jth body panel control point. These are used to form the nonuniform flow boundary condition on the store excluding any image store effects.
YIM(J),ZIM(J)	y,z coordinates of Jth image store control point in original source panel coordinate system
SVN(J), LSKP(J)	work arrays for checking vanishing influence at Jth control point
AN(I,J)	influence coefficient matrix containing influence of panels in one ring on a second ring. Used in panel strength and velocity calculations.

NBODY is the number of source panels on the separated store and KRAD is the number of panels in a ring around the body.

The third set of temporary arrays is defined in routine DEMON2 for use in computing the influence of the body source panels at empennage control points. This operation is performed at the point indicated as item 7 in Figure C-2. The calculations

performed here complement those performed in BDCOEF by using the arrays defined there to compute velocities at the control point. Only one array is allocated space in blank common. During these calculations blank common is equivalently dimensioned as

```
COMMON   A(NTAP7), UB(3,NFLD,MAXKR)
```

where the above array contains the following information

UB(I,J,K)	coefficient array to hold u,v,w velocity components for influence of Kth source panel in a ring at Jth empennage control point

The empennage control points are those specified in the definition of the coefficients in BDCOEF. The coefficients for one ring on all control points are read from TAPE10 as needed.

The third use of blank common is for temporary storage of the empennage influence coefficient matrices during generation and solution for the constant u-velocity strengths. The influence coefficient matrix, FVN, is generated in routine CRFWBD and stored on TAPE3 at the point indicated as item 4 in Figure C-2. Similarly, it is retrieved in routine DEMON2 from TAPE3 at the point indicated as item 8 in Figure C-2. The array is read immediately prior to its use. Until that point, the arrays for the noncircular bodies are required in blank common for the calculation of the body influences at empennage control points. The FVN array is only required in core long enough to solve the system of equations for the empennage panel strengths.

When multiple empennages are being analyzed, the body data must be brought back in common during the analysis for the influence on the second empennage. The second FVN array is then copied into blank common in the same location as the previous one. In addition, the arrays saved in EGMSAV on TAPE3 are retrieved by EGMRST at item 6 for each empennage and restored to their

appropriate labeled commons. This sequence is repeated for each integration step in the trajectory calculation.

After the points indicated as items 4 and 8 in Figure C-2, blank common is equivalently dimensioned as:

```
COMMON    FVN(62500)
```

It is noted that the size of the actual array sets the dimension requirements for blank common. The current dimension limit above allows for 250 constant u-velocity panels to be placed on each of two empennages.

TABLE D-1

Multiple Configuration Blank Common Variable
Locations - Separating Store

(Separating store data offset index: IDEJ)

No.	Name	Length	1st location from offset	1st address in blank common
1	XPT	NBODY	IXPT	A(IDEJ+IXPT)
2	YPT	NBODY	IYPT	A(IDEJ+IYPT)
3	ZPT	NBODY	IZPT	A(IDEJ+IZPT)
4	THET	NOBDY	ITH	A(IDEJ+ITH)
5	DELTA	NBODY	IDEL	A(IDEJ+IDEL)
6	AREA	NBODY	IAR	A(IDEJ+IAR)
7	XC	KX	IXC(1)	A(IDEJ+IXC(1))
8	YC	KXKR	IYC(1)	A(IDEJ+IYC(1))
9	ZC	KXKR	IZC(1)	A(IDEJ+IZC(1))
10	GB	NBODY*ISHPEJ	IGB(NEJGB)	A(IDEJ+IGB(NEJGB))
11	U	NBODY	IU	A(IDEJ+IU)
12	V	NBODY	IV	A(IDEJ+IV)
13	W	NBODY	IW	A(IDEJ+IW)
14	CP	NBODY	ICP	A(IDEJ+ICP)

Note: IDEJ=0 in Program II

TABLE D-2

Multiple Configuration Blank Common Variable
Locations - Fuselage

(Fuselage data offset index: IDF)

No.	Name	Length	1st location from offset	1st address in blank common
1	NX	1	1	IA(IDF+1)
2	NR	1	2	IA(IDF+2)
3	KX	1	3	IA(IDF+3)
4	KR	1	4	IA(IDF+4)
5	NXNR		5	IA(IDF+5)
6	KXKR		6	IA(IDF+6)
7	NXKR		7	IA(IDF+7)
8	MAXNX		8	IA(IDF+8)
9	MAXKR		9	IA(IDF+9)
10	NATOT		10	IA(IDF+10)
11	NBODY		11	IA(IDF+11)
12	KFUS	1	12	IA(IDF+12)
13	KRADX	5	13	IA(IDF+13)
14	KFORX	5	18	IA(IDF+18)
15	IXC	5	23	IA(IDF+23)
16	IYC	5	28	IA(IDF+28)
17	IZC	5	33	IA(IDF+33)
18	IXZSYM	1	38	IA(IDF+38)
19	NADIM		39	IA(IDF+39)
20	NXDIM		40	IA(IDF+40)
21	NG		41	IA(IDF+41)
22	IXPT		42	IA(IDF+42)
23	IYPT		43	IA(IDF+43)
24	IZPT		44	IA(IDF+44)
25	ITH		45	IA(IDF+45)
26	IDEL		46	IA(IDF+46)
27	NTAP7		47	IA(IDF+47)
28	IAR		48	IA(IDF+48)
29	IAN		49	IA(IDF+49)
30	IUB	1	50	IA(IDF+50)
31	IGB	7	51	IA(IDF+51)
32	IVB	1	58	IA(IDF+58)
33	IU		59	IA(IDF+59)
34	IV		60	IA(IDF+60)
35	IW		61	IA(IDF+61)
36	IVA		62	IA(IDF+62)
37	IWA		63	IA(IDF+63)
38	ICP		64	IA(IDF+64)
39	IPHI		65	IA(IDF+65)
40	IYB	1	66	IA(IDF+66)
41	NAG	1	67	IA(IDF+67)
42	NAP	1	68	IA(IDF+68)

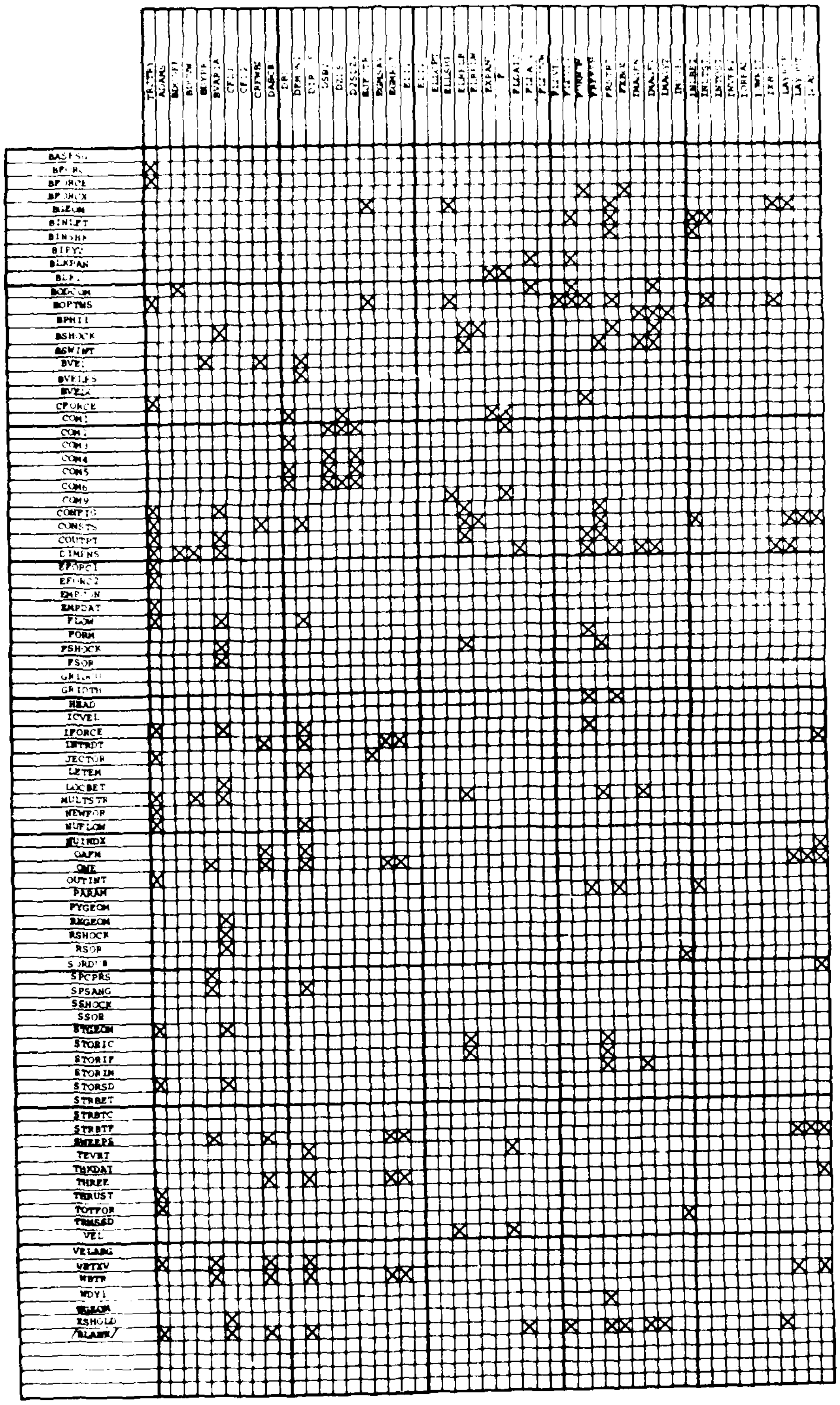
TABLE D-2 (cont.)

No.	Name	Length	1st location from offset	1st address in blank common
43	NAV	1	69	IA(IDF+69)
44	NAS		70	IA(IDF+70)
45	NASHK		71	IA(IDF+71)
46	NAFLD		72	IA(IDF+72)
47	IA0		73	IA(IDF+73)
48	ID0		74	IA(IDF+74)
49	ISK0	1	75	IA(IDF+75)
50	IZ2	4	76	IA(IDF+76)
51	NRING	1	80	IA(IDF+80)
52	IROW	50	81	IA(IDF+81)
53	XMACH	1	131	A(IDF+131)
54	ALPHA		132	A(IDF+132)
55	BETA		133	A(IDF+133)
56	ALPHAC		134	A(IDF+134)
57	PHIR		135	A(IDF+135)
58	EM		136	A(IDF+136)
59	SINAC		137	A(IDF+137)
60	COSAC		138	A(IDF+138)
61	SPHI		139	A(IDF+139)
62	CPHI		140	A(IDF+140)
63	SINA		141	A(IDF+141)
64	SINB	1	142	A(IDF+142)
65	J0	1	143	IA(IDF+143)
66	J2		144	IA(IDF+144)
67	J6		145	IA(IDF+145)
68	NFUS	1	146	IA(IDF+146)
69	NRADX	5	147	IA(IDF+147)
70	NFORX	5	152	IA(IDF+152)
71	J2TEST	1	157	IA(IDF+157)
72	IPRES	1	158	IA(IDF+158)
73	ISOLV	1	159	IA(IDF+159)
74	INLET	1	160	LA(IDF+160)
75	IPLOT	4	161	IA(IDF+161)
76	IPRT	5	165	IA(IDF+165)
77	IUVW	1	170	IA(IDF+170)
78	XSTART		171	A(IDF+171)
79	XWLE		172	A(IDF+172)
80	REFA		173	A(IDF+173)
81	REFD		174	A(IDF+174)
82	REFL		175	A(IDF+175)
83	REFX		176	A(IDF+176)
84	REFZ		177	A(IDF+177)
85	CCTEST		178	A(IDF+178)
86	ITMAX	1	179	IA(IDF+179)
87	BODL	1	180	A(IDF+180)
88	IZ1	12	181	IA(IDF+181)

TABLE D-2 (conc.)

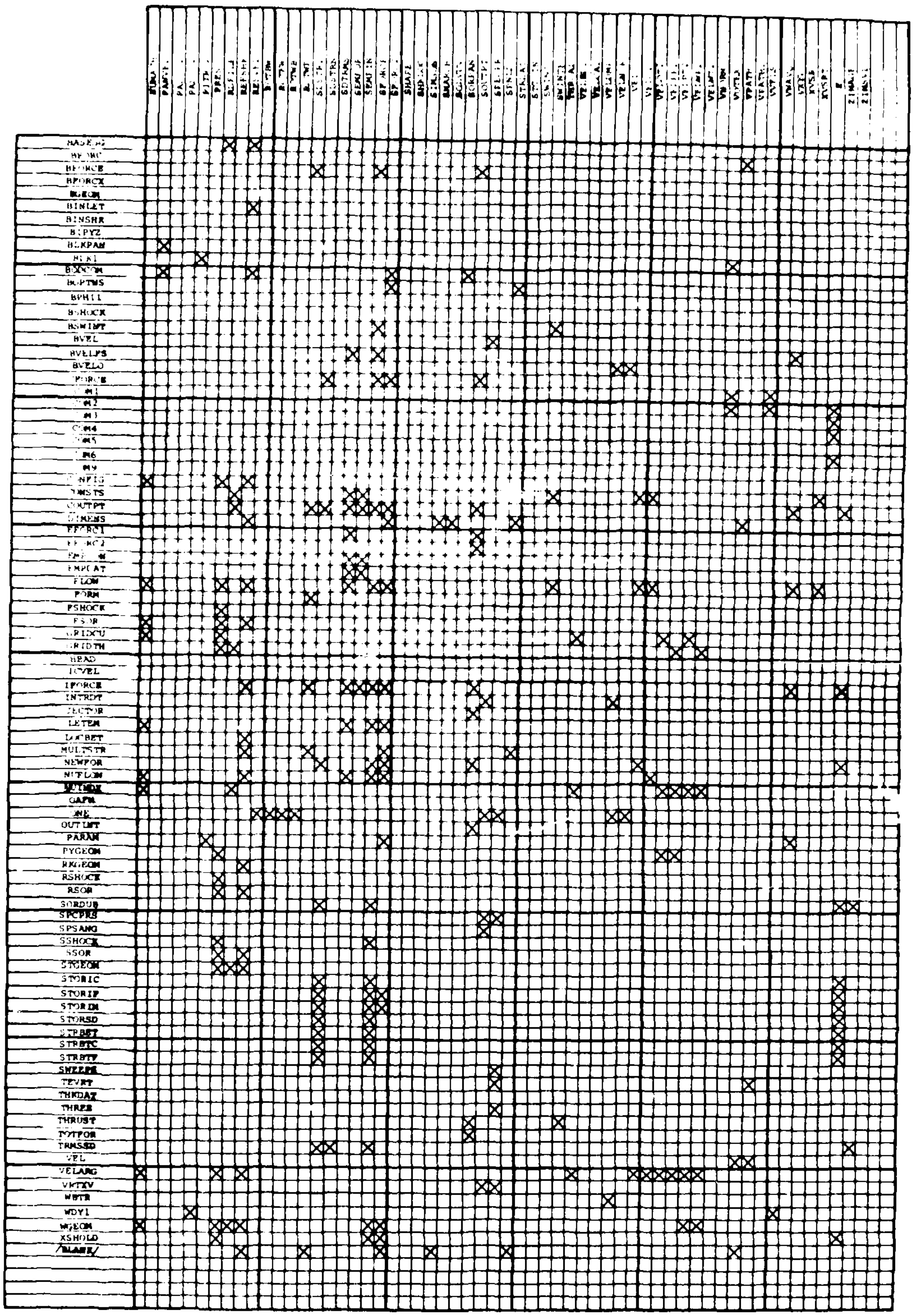
No.	Name	Length	1st location from offset	1st address in blank common
89	XFUS	51	193	A(IDF+193)
90	ZFUS	51	244	A(IDF+244)
91	FUSARD	51	295	A(IDF+295)
92	FUSBY	51	346	A(IDF+346)
93	FUSAZ	51	397	A(IDF+397)
94	XJ	51	448	A(IDF+448)
95	PHIK	33	499	A(IDF+499)
96	TITLE1	20	532	A(IDF+532)
97	TITLE2	20	552	A(IDF+552)
98	NSHK	10	572	IA(IDF+572)
99	PHIS	10	582	A(IDF+582)
100	THETN	10	592	A(IDF+592)
101	MAXSHK	1	602	IA(IDF+602)
102	NSHOCK	1	603	IA(IDF+603)
103	DBETA	1	604	A(IDF+604)
104	EALPHA	1	605	A(IDF+605)
105	CNU0	1	606	A(IDF+606)
106	CNU2	1	607	A(IDF+607)
107	XSHLDR	1	608	A(IDF+608)
108	SHK	3	609	A(IDF+609)
109	XSHK	100	612	A(IDF+612)
110	RSHK	100	712	A(IDF+712)
111	XPT	NBODY	IXPT+811	A(IDF+811+IXPT)
112	YPT	NBODY	IYPT+811	A(IDF+811+IYPT)
113	ZPT	NBODY	IZPT+811	A(IDF+811+IZPT)
114	THET	NBODY	ITH+811	A(IDF+811+IPT)
115	DELTA	NBODY	IDEL+811	A(IDF+811+IDEL)
116	AREA	NBODY	IAR+811	A(IDF+811+IAR)
117	XC	KX	IXC(1)+811	A(IDF+811+IXC(1))
118	YC	KXKR	IYC(1)+811	A(IDF+811+IYC(1))
119	ZC	KXKR	IZC(1)+811	A(IDF+811+IZC(1))
120	GB	NBODY	IGB(1)+811	A(IDF+811+IGB(1))

Notes:
(1) ID0 = IDF, both may be used interchangeably
(2) ISK0 = IDF+571, it may be used as offset for items 98 through 110
(3) IA0 = IDF+811, it may be used as offset for items 111 through 120
(4) To obtain locations of second elliptic store shape substitute ID2 for IDF in all addresses above



(a)

Figure D-1.- Common statements and routines in Program II in which they appear.



(b)

Figure D-1.- Concluded.

REFERENCES

1. Mullen, J., Jr., Goodwin, F. K., Dillenius, M. F. E.: Prediction of Supersonic Store Separation Characteristics Including Fuselage and Stores of Noncircular Cross Section. Vol. II - User's Manual for the Program. Technical Report AFWAL-TR-80-3032, Vol. II, November 1980.

2. Goodwin, F. K., Dillenius, M. F. E. and Mullen, J., Jr.: Prediction of Supersonic Store Separation Characteristics Including Fuselage and Stores of Noncircular Cross Section. Vol. I - Theoretical Methods and Comparisons with Experiment. Technical Report AFWAL-TR-80-3032, Vol. I, November 1980.

3. Hildebrand, F. B.: Introduction to Numerical Analysis. McGraw-Hill Book Co., New York, NY, 1956, Chapter 6.

4. IBM System/360 Scientific Subroutine Package, Version III, Programmers Manual. IBM Corp., GH20-0205-4, August 1970, pp. 372-381.

5. Dillenius, M. F. E. and Nielsen, J. N.: Computer Programs for Calculating Pressure Distributions Including Vortex Effects on Supersonic Monoplane or Cruciform Wing-Body-Tail Combinations with Round or Elliptical Bodies. NASA CR-3122, April 1979.

6. Goodwin, F. K., Dillenius, M. F. E., and Nielsen, J. N.: Prediction of Six-Degree-of-Freedom Store Separation Trajectories at Speeds up to the Critical Speed. Vol. I - Theoretical Methods and Comparison with Experiment. Technical Report AFFDL-TR-72-83, Vol. I, October 1974.

7. Scarborough, J. B.: Numerical Mathematical Analysis. Johns Hopkins Press, Baltimore, 1966, p. 137.

8. Woodward, F. A.: An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow, Part II - Computer Program Description. NASA CR-2228, Part II, May 1973.

9. Dillenius, M. F. E., Goodwin, F. K., and Nielsen, J. N.: Prediction of Supersonic Store Separation Characteristics. Vol. 1 - Theoretical Methods and Comparisons with Experiment. Technical Report AFFDL-TR-76-41, Vol. I, May 1976.

☆U.S.Government Printing Office: 1981 — 757-002/462

LMED

8